



**TUGAS AKHIR TF 145565**

**RANCANG BANGUN SISTEM *MONITORING*  
KONSENTRASI GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
SEBAGAI EMISI GAS BUANG MENGGUNAKAN  
SENSOR MG 811 BERBASIS STM32F4 *DISCOVERY***

**Syahril Arisdianta  
NRP. 2414 031 017**

**Dosen Pembimbing  
Dr. Ir. Ali Musyafa', M.Sc.  
NIP. 19600901 198701 1 001**

**PROGRAM STUDI D3 TEKNIK INSTRUMENTASI  
DEPARTEMEN TEKNIK INSTRUMENTASI  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017**



**TUGAS AKHIR – TF 145565**

**RANCANG BANGUN SISTEM *MONITORING*  
KONSENTRASI GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
SEBAGAI EMISI GAS BUANG MENGGUNAKAN  
SENSOR MG 811 BERBASIS STM32F4 *DISCOVERY***

Syahril Arisdianta  
NRP. 2414 031 017

Dosen Pembimbing  
Dr. Ir. Ali Musyafa', M.Sc.  
NIP. 19600901 198701 1 001

PROGRAM STUDI D3 TEKNIK INSTRUMENTASI  
DEPARTEMEN TEKNIK INSTRUMENTASI  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017



***FINAL PROJECT - TF 095565***

***DESIGN SYSTEM MONITORING CONCENTRATION SYSTEM  
GAS CARBON DIOXIDE (CO<sub>2</sub>) AS EMISSION GAS WASTE  
USING MG 811 SENSOR BASED STM32F4 DISCOVERY***

Syahril Arisdianta  
NRP. 2414 031 017

*Advisor Lecturer*  
Dr. Ir. Ali Musyafa', M. Sc.  
NIP. 19600901 198701 1 001

***D3 INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF VOCATION  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017***

**RANCANG BANGUN SISTEM *MONITORING*  
KONSENTRASI GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
SEBAGAI EMISI GAS BUANG MENGGUNAKAN  
SENSOR MG 811 BERBASIS STM32F4 *DISCOVERY***

**TUGAS AKHIR**

**Oleh :**

**SYAHRIL ARISDIANTA**  
**NRP. 2414 031 017**

**Surabaya, 25 Juli 2017**  
**Mengetahui / Menyetujui**

**Dosen Pembimbing**

**Dr. Ir. Ali Musyafa', M.Sc.**  
**NIP. 19600901 198701 1 001**



**RANCANG BANGUN SISTEM *MONITORING*  
KONSENTRASI GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
SEBAGAI EMISI GAS BUANG MENGGUNAKAN  
SENSOR MG 811 BERBASIS STM32F4 *DISCOVERY***

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Ahli Madya  
Pada  
Program Studi D3 Teknik Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Oleh :  
**SYAHRIL ARISDIANTA**  
NRP. 2414 031 017

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Ali Musyafa', M.Sc.. ..... (Dosen Pembimbing)
2. Dr. Ir. Purwadi Agus D., M.Sc ..... (Ketua Tim Penguji)
3. Detak Yan Pratama S.T., M. Sc. .... (Dosen Penguji 1)

**SURABAYA  
JULI 2017**

**RANCANG BANGUN SISTEM *MONITORING*  
KONSENTRASI GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
SEBAGAI EMISI GAS BUANG MENGGUNAKAN  
SENSOR MG 811 BERBASIS STM32F4 *DISCOVERY***

**Nama Mahasiswa** : Syahril Arisdianta  
**NRP** : 2414 031 017  
**Program Studi** : D III Teknik Instrumentasi  
**Jurusan** : Teknik Instrumentasi FV-ITS  
**Dosen Pembimbing** : Dr. Ir. Ali Musyafa', M.Sc.

**Abstrak**

Pencemaran udara adalah suatu kondisi di mana kualitas udara menjadi rusak dan terkontaminasi oleh zat-zat, baik yang tidak berbahaya maupun yang membahayakan kesehatan tubuh manusia dan dengan adanya karbon dioksida (CO<sub>2</sub>) yang berlebih di udara dapat mengurangi kesegaran dan kebersihan udara yang kita hirup. Karbon dioksida (CO<sub>2</sub>) juga bias menjadi polusi udara apabila kadarnya dalam udara berlebih, dapat mengakibatkan gangguan pernapasan juga keracunan terhadap susunan saraf. Bahkan dalam kadar tertentu akan membunuh makhluk hidup yang ada di bumi. Oleh sebab itu diperlukannya alat monitoring emisi gas karbon dioksida (CO<sub>2</sub>) berbasis mikrokontroler. Mikrokontroler STM32F4 Discovery digunakan untuk memproses sinyal keluaran dari modul sensor karbon dioksida MG 811. Berdasarkan analisa data, dapat diketahui faktor cakupan sebesar 1.826 diperoleh dari T Student dengan tingkat kepercayaan 95% dan alat ukur memiliki nilai ketidakpastian diperluas ( $U_{exp}$ ) sebesar  $\pm 7,502121$  dengan tingkat kepercayaan 95% sehingga diperoleh hasil pengukuran  $\bar{x} = (522,3 \pm 7,502121)$ ppm.

**Kata kunci:** Karbon Dioksida (CO<sub>2</sub>), Sensor MG 811, STM32F4 *DISCOVERY*, Standar Deviasi

***DESIGN SYSTEM MONITORING CONCENTRATION  
SYSTEM GAS CARBON DIOXIDE (CO<sub>2</sub>) AS EMISSION  
GAS WASTE USING MG 811 SENSOR BASED STM32F4  
DISCOVERY***

***Name of Student*** : Syahril Arisadiana  
***NRP*** : 2414 031 017  
***Program Study*** : D III Instrumentation Engineering  
***Department*** : Instrumentation Engineering FV-ITS  
***Advisor Lecturer*** : Dr. Ir. Ali Musyafa', M. Sc.

***ABSTRACT***

*Air pollution is a condition in which air quality becomes damaged and contaminated by substances, both harmless and harmful to the health of the human body and the presence of excess carbon dioxide (CO<sub>2</sub>) in the air can reduce the freshness and cleanliness of the air we breathe. Carbon dioxide (CO<sub>2</sub>) is also biased into air pollution when the levels are in excess air, can cause respiratory distress also toxicity to the nervous system. Even in a certain degree will kill the living creatures that exist on earth. Therefore the need for monitoring tool emission of carbon dioxide gas (CO<sub>2</sub>) based on microcontroller. The STM32F4 Discovery Microcontroller is used to process the output signal from the MG 811 carbon dioxide sensor module. Based on the data analysis, it can be seen the coverage factor of 1.826 is obtained from Student T with 95% confidence level and the measuring instrument has an expanded uncertainty value  $U_{exp}$  equal to  $\pm 7,502121$  with 95% confidence level so that obtained result of measurement  $x = (522,3 \pm 7,502121)\text{ppm}$ .*

***Keywords: Carbon Dioxide (CO<sub>2</sub>), MG 811 Sensor, STM32F4  
DISCOVERY, Standard Deviation***

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas limpahan rahmat dan hidayah-Nya serta shalawat dan salam kepada Nabi Muhammad SAW sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Sistem *Monitoring* Konsentrasi Gas Karbon Dioksida (CO<sub>2</sub>) Sebagai Emisi Gas Buang Menggunakan Sensor Mg 811 Berbasis STM32F4 *Discovery*”**. Penulis telah banyak mendapatkan bantuan dari berbagai pihak dalam menyelesaikan Tugas Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc yang telah sabar memberikan dukungan, bimbingan, serta ilmu yang sangat bermanfaat.
2. Bapak Dr. Ir. Ali Musyafa', M.Sc selaku dosen pembimbing yang senantiasa memberikan motivasi, bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini.
3. Bapak dan ibu penguji Tugas Akhir yang telah banyak memberikan masukan dan arahan dalam menyelesaikan Tugas Akhir ini.
4. Bapak dan ibu kepala Laboratorium di Teknik Fisika yang telah memberikan sarana dan prasarana guna menunjang pelaksanaan Tugas Akhir ini.
5. Bapak dan Ibu dosen Teknik Instrumentasi yang telah memberikan ilmu selama kuliah.
6. Seluruh Staf Jurusan Teknik Instrumentasi yang telah membantu penulis dalam hal administrasi.
7. Kedua orang tua dan seluruh keluarga besar tercinta yang senantiasa memberikan dukungan, semangat dan do'a kepada penulis.
8. Teman-teman seperjuangan dalam mengerjakan Tugas Akhir Tim *monitoring concentration system gas* (Lailatul Mufida, Atik Sinawang, dan Haryo Arif Wicaksono), Teman-teman D3 Teknik Instrumentasi lainnya serta



teman-teman S1 Teknik Fisika angkatan 2014 yang selalu memotivasi penulis.

Penulis menyadari bahwa penulisan laporan Tugas Akhir ini tidaklah sempurna. Oleh karena itu sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik lagi. Penulis juga berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembacanya.

Surabaya, Juli 2017

Penulis.

## DAFTAR ISI

<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b><i>TITLE OF PAGE</i> .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN I.....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN II.....</b>	<b>iv</b>
<b>ABSTRAK.....</b>	<b>v</b>
<b><i>ABSTRACT</i> .....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL.....</b>	<b>xiv</b>

### **BAB I PENDAHULUAN**

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah .....	3
1.5 Manfaat.....	3
1.6 Sistematika Laporan .....	4

### **BAB II TINJAUAN PUSTAKA**

2.1 <i>Monitoring</i> .....	5
2.2 Karbon Dioksida.....	6
2.3 Motor DC.....	7
2.4 STM32F4 Discovery .....	8
2.5 Sensor MG 811 .....	10
2.6 LCD 16 x 4 .....	12
2.7 <i>Micro Sd shield module</i> .....	13
2.8 RTC DS3231 .....	15
2.9 <i>Modem GSM Wavecom Fastrack M1306B</i> .....	16
2.10 <i>MCB (Miniature Circuit Breaker)</i> .....	18
2.11 <i>ChibiOS</i> .....	19
2.12 Karakteristik Statik .....	20
2.13 Teori Ketidakpastian.....	21
2.14 Konfigurasi <i>Port Serial</i> .....	24

2.15 <i>Qt Qreators</i> .....	25
2.16 <i>MicroSD</i> .....	27
2.17 <i>Notepad++</i> .....	28

## **BAB III PERANCANGAN DAN PEMBUATAN ALAT**

3.1 <i>Flowchart</i> dan Diagram Blok Perancangan Alat.....	31
3.2 Rangkaian <i>Power supply</i> .....	34
3.3 Perancangan Sensor MG-811 .....	35
3.4 Perancangan <i>Real Time Clock (RTC)</i> .....	37
3.5 Perancangan <i>Data Logger</i> .....	38
3.6 Perancangan <i>Modem Wavcom</i> .....	39
3.7 <i>PCB Designer</i> .....	41
3.8 Rangkaian <i>Power Supply</i> .....	42
3.9 Perancangan dan Pembuatan <i>Software</i> .....	43

## **BAB IV PENGUJIAN DAN ANALISIS DATA**

4.1 Pengujian Alat .....	53
4.2 Pengujian Penyimpan Memori ke SD Cards .....	54
4.3 Pengujian Modem untuk SMS Gateway.....	55
4.4 Pengujian <i>RTC (Real Time Clock)</i> .....	57
4.5 Pengujian Sensor Gas MG 811 .....	59
4.6 Data Spesifikasi Alat .....	63
4.6 Pembahasan .....	69

## **BAB V PENUTUP**

5.1 Kesimpulan.....	71
5.2 Saran .....	71

## **DAFTAR PUSTAKA**

**LAMPIRAN A (*LISTING PROGRAM PADA CHIBIOS/RT*)**

**LAMPIRAN B (*DATA SHEET STM32F4 DISCOVERY*)**

**LAMPIRAN C (*DATA SHEET SENSOR MG 811*)**

**LAMPIRAN D (*LAPORAN HASIL PENGAMBILAN DATA*)**

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	<b>Diagram Blok <i>Monitoring</i></b> .....	5
<b>Gambar 2.2</b>	<b>DC <i>Exhaust Fan</i></b> .....	8
<b>Gambar 2.3</b>	<b>STM32F4 <i>Discovery</i></b> .....	8
<b>Gambar 2.4</b>	<b>Bentuk Fisik MG-811</b> .....	10
<b>Gambar 2.5</b>	<b>Struktur Sensor MG-811</b> .....	11
<b>Gambar 2.6</b>	<b>Bagian Dalam Sensor MG-811</b> .....	11
<b>Gambar 2.7</b>	<b>LCD 16x4</b> .....	12
<b>Gambar 2.8</b>	<b><i>modul Micro Sd (1)</i></b> .....	14
<b>Gambar 2.9</b>	<b><i>modul Micro Sd (2)</i></b> .....	14
<b>Gambar 2.10</b>	<b>RTC (<i>Real Time Clock</i>)</b> .....	15
<b>Gambar 2.11</b>	<b><i>Modem Wavecom Fastrack M1306B Serial</i></b> .....	17
<b>Gambar 2.12</b>	<b><i>Arsitektur Modem Wavecom Fastrack M1306B Serial</i></b> .....	18
<b>Gambar 2.13</b>	<b>MCB Shukaku 220 V 2A</b> .....	19
<b>Gambar 2.14</b>	<b>Logo ChibiOS</b> .....	19
<b>Gambar 2.15</b>	<b>Tabel <i>T-student</i></b> .....	23
<b>Gambar 2.16</b>	<b><i>Port Serial</i></b> .....	24
<b>Gambar 2.17</b>	<b>Logo <i>QT Creator</i></b> .....	26
<b>Gambar 2.18</b>	<b>Logo <i>microSD</i></b> .....	27
<b>Gambar 2.19</b>	<b>Logo <i>Notepad++</i></b> .....	28
<b>Gambar 3.1</b>	<b><i>Flowchart Pengerjaan Tugas Akhir</i></b> .....	31
<b>Gambar 3.2</b>	<b>Diagram Blok Sistem monitoring gas buang</b> .....	33
<b>Gambar 3.3</b>	<b>Skematik <i>Plant</i> monitoring gas</b> .....	34
<b>Gambar 3.4</b>	<b><i>Power Supply</i> DC</b> .....	35
<b>Gambar 3.5</b>	<b>Blok Diagram <i>Power Supply</i></b> .....	35
<b>Gambar 3.6</b>	<b>Karbon dioksida(CO<sub>2</sub>) Gas Sensor Module</b> .....	36
<b>Gambar 3.7</b>	<b>Blok Diagram <i>Sensor MG 811</i></b> .....	36
<b>Gambar 3.8</b>	<b>Skematik RTC pada ARM</b> .....	38
<b>Gambar 3.9</b>	<b>Skematik SD card pada ARM</b> .....	39
<b>Gambar 3.10</b>	<b>Konfigurasi port modem wavecom</b> .....	40
<b>Gambar 3.11</b>	<b><i>Layout PCB Designer</i></b> .....	41
<b>Gambar 3.12</b>	<b>Rangkaian LCD pada STM32F4 <i>Discovery</i></b> .....	42
<b>Gambar 3.13</b>	<b>Penempatan LCD untuk <i>Display</i></b> .....	43

<b>Gambar 3.14</b> Rangkaian Skematik Mikrokontroler STM32F4, sensor MG 811, SD Card, RTC, Uart (SMS Gateway) dan LCD 16x4.....	43
<b>Gambar 3.15</b> <i>Create New Project</i> .....	44
<b>Gambar 3.16</b> <i>Project Qt Creator</i> .....	44
<b>Gambar 3.17</b> <i>import existing project name and location</i> .....	45
<b>Gambar 3.18</b> <i>import existing project file selection</i> .....	46
<b>Gambar 3.19</b> <i>import existing project management</i> .....	47
<b>Gambar 3.20</b> <i>Tampilan awal program</i> .....	48
<b>Gambar 3.21</b> <i>Program project .files</i> .....	48
<b>Gambar 3.22</b> <i>Program project .include</i> .....	49
<b>Gambar 3.23</b> <i>Class pada project</i> .....	49
<b>Gambar 3.24</b> <i>Build Project Qt Creator</i> .....	50
<b>Gambar 3.25</b> <i>Make all project di Notepad++</i> .....	51
<b>Gambar 3.26</b> <i>Download project di ST-LINK V2</i> .....	51
<b>Gambar 4.1</b> Alur Sistem monitoring konsentrasi gas sebagai emisi gas buang .....	53
<b>Gambar 4.2</b> <i>Plant monitoring gas</i> .....	54
<b>Gambar 4.3</b> Pengujian SD Card.....	55
<b>Gambar 4.4</b> <i>Screenshoot AT Command pada Hyperterminal</i> .....	56
<b>Gambar 4.5</b> <i>Screenshoot SMS yang diterima pada HP</i> .....	56
<b>Gambar 4.6</b> <i>Screenshoot SMS Data Hasil Monitoring</i> .....	57
<b>Gambar 4.7</b> Pengujian data waktu RTC pada Hyperterminal..	58
<b>Gambar 4.8</b> Pengujian data waktu .....	58
<b>Gambar 4.9</b> Peletakan Sensor CO <sub>2</sub> .....	59
<b>Gambar 4.10</b> Grafik Pembacaan Standar dan Pembacaan Alat 60	
<b>Gambar 4.11</b> Grafik Pembacaan Standar dan Pembacaan ADC Alat .....	61
<b>Gambar 4.12</b> Pengujian SD Card Grafik Pembacaan Standar dan Pembacaan Alat dengan persamaan matematik .....	62

## DAFTAR TABEL

<b>Tabel 2.1</b> Pengaruh Gas CO <sub>2</sub> terhadap Kesehatan.....	7
<b>Tabel 2.2</b> Spesifikasi Sensor MG 811 .....	11
<b>Tabel 2.3</b> Pin Konfigurasi LCD 16x4.....	12
<b>Tabel 4.1</b> Data Pengujian Alat.....	60
<b>Tabel 4.2</b> Tabel Konversi Data ADC ke PPM.....	62
<b>Tabel 4.3</b> Pengambilan Data pada <i>Sensor</i> MG 811 .....	63
<b>Tabel 4.4</b> Data Kalibrasi pada <i>Sensor</i> MG 811 (1).....	65
<b>Tabel 4.5</b> Data Kalibrasi pada <i>Sensor</i> MG 811 (2).....	66

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Pencemaran udara adalah suatu kondisi di mana kualitas udara menjadi rusak dan terkontaminasi oleh zat-zat, baik yang tidak berbahaya maupun yang membahayakan kesehatan tubuh manusia. Pencemaran udara biasanya terjadi di kota-kota besar dan juga daerah padat industri yang menghasilkan gas-gas yang mengandung zat di atas batas kewajaran. Rusaknya atau semakin sempitnya lahan hijau atau pepohonan di suatu daerah juga dapat memperburuk kualitas udara di tempat tersebut. Semakin banyak kendaraan bermotor dan alat-alat industri yang mengeluarkan gas yang mencemarkan lingkungan akan semakin parah pula pencemaran udara yang terjadi. Untuk itu diperlukan peran serta pemerintah, pengusaha dan masyarakat untuk dapat menyelesaikan permasalahan pencemaran udara yang terjadi.<sup>[1]</sup>

Karbon dioksida adalah zat gas yang mampu meningkatkan suhu pada suatu lingkungan sekitar kita yang disebut juga sebagai efek rumah kaca. Dengan begitu maka temperatur udara di daerah yang tercemar CO<sub>2</sub> itu akan naik dan otomatis suhunya menjadi semakin panas dari waktu ke waktu seperti di wilayah DKI Jakarta. Hal ini disebabkan karena CO<sub>2</sub> akan berkonsentrasi dengan jasad renik, debu dan titik-titik air yang membentuk awan yang dapat ditembus cahaya matahari namun tidak dapat melepaskan panas ke luar awan tersebut. Keadaan seperti itu mirip dengan kondisi rumah kaca tanpa AC dan ventilasi udara yang cukup.<sup>[2]</sup>

Menurut Kepala Badan Lingkungan Hidup (BLH) Provinsi Jawa Timur, Dewi J Putriatni, menyebutkan, Surabaya menduduki peringkat ketiga setelah Bangkok dan Jakarta sebagai kota di kawasan Asia yang polusi udaranya buruk. Dalam memantau kondisi tingkat polusi udara yang ada di wilayah Surabaya, Pemerintah Kota Surabaya juga telah memiliki sarana dan prasarana berupa ISPU (Indeks Standar Pencemaran Udara). Akan tetapi, hanya ada 2 display ISPU saja yang masih berfungsi saat ini, yakni di jalur MERR dan satu lagi didepan hotel Sahid. Sedangkan

4 display ISPU yang lain yakni di jalan meyjen sungkono, gubeng, kertjaya, dan jalan pahlawan dalam kondisi tidak rusak<sup>[3]</sup>.

Maka dibuat alat yang berfungsi memonitoring kadar gas karbon dioksida ( $\text{CO}_2$ ) di udara. Alat ini juga akan bersifat portabel sehingga mempermudah dalam penggunaannya. Ketika proses pengukuran kadar karbon dioksida ( $\text{CO}_2$ ) berlangsung, terjadi perubahan kadar karbon dioksida ( $\text{CO}_2$ ) yang dapat dideteksi oleh sensor gas karbon dioksida ( $\text{CO}_2$ ). Perubahan akan masuk ke dalam rangkaian. Kemudian diproses menggunakan mikrokontroler untuk dihitung konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) yang diukur. Penggunaan mikrokontroler didasarkan pada kemudahan dalam pemrosesan data karena bahasa C-nya relatif mudah dan mikrokontroler memiliki fungsi yang dapat diterapkan dalam alat monitoring kadar gas karbon dioksida ( $\text{CO}_2$ ). Alat ini akan menampilkan kadar gas karbon dioksida ( $\text{CO}_2$ ) di udara pada display LCD. Oleh sebab itu dilakukan pembuatan dan perancangan alat monitoring emisi gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang berbasis mikrokontroler STM32F4 Discovery yang mudah digunakan sehingga pengukuran terhadap emisi gas dapat dilakukan dengan mudah, dan penerapan *standart* emisi gas dapat dilakukan dengan baik.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

- a. Bagaimana merancang suatu sistem monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang berbasis mikrokontroler STM32F4 Discovery?
- b. Bagaimana hasil dari kalibrasi suatu sistem monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang berbasis mikrokontroler STM32F4 Discovery?
- c. Bagaimana penggunaan data hasil monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang?



### 1.3 Tujuan

Tujuan yang dicapai dalam tugas akhir ini adalah sebagai berikut:

- a. Merancang suatu sistem monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang berbasis mikrokontroler STM32F4 Discovery.
- b. Mengetahui hasil dari kalibrasi suatu sistem monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang berbasis mikrokontroler STM32F4 Discovery?
- c. Mengetahui penggunaan data hasil monitoring kandungan gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang.

### 1.4 Batasan Masalah

Perlu diberikan beberapa batasan masalah agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan masalah dari sistem yang dirancang ini adalah sebagai berikut:

- a. Merancang sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sebuah mikrokontroler STM32F4 *Discovery*.
- b. Merancang sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sensor MG-811.
- c. Merancang sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sebuah *display* LCD ukuran 16x4 karakter.
- d. Pengujian sistem dari rancang bangun yang telah dibuat dengan menguji performansi alat, baik keakuratan dan keoptimalan alat.
- e. Menyusun hasil teori dari pembuatan *hardware*, analisa data dan kesimpulan dari data dan sistem yang ada.

### 1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan mikrokontroler STM32F4 Discovery dan dapat dijadikan sebagai perancangan alat monitoring ISPU (Indeks Standar Pencemaran Udara) masa depan..

## **1.6 Sistematika Laporan**

Sistematika laporan yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, dan sistematika penulisan dalam tugas akhir ini.

### **BAB II TINJAUAN PUSTAKA**

Pada bab ini membahas mengenai teori-teori penunjang yang diperlukan dalam merealisasikan tugas akhir yaitu berupa teori tentang monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ), dan perangkat-perangkat yang digunakan dalam pembuatan tugas akhir ini.

### **BAB III PERANCANGAN DAN PEMBUATAN ALAT**

Pada bab ini diuraikan tentang penjelasan mengenai perancangan dan pembuatan alat.

### **BAB IV PENGUJIAN DAN ANALISIS DATA**

Pada bab ini memuat tentang hasil pengujian dari perangkat yang dibuat beserta pembahasannya.

### **BAB V PENUTUP**

Pada bab ini memuat tentang kesimpulan dan saran dari pembuatan tugas akhir ini.

## BAB II

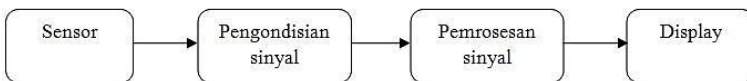
### TINJAUAN PUSTAKA

#### 2.1 *Monitoring*

*Monitoring* merupakan pengawasan pada suatu variabel atau sistem yang bertujuan untuk mengamati keadaan secara *realtime*. *Monitoring* dilakukan untuk mendeteksi jika akan terjadi suatu kegagalan atau gangguan pada sistem sehingga dapat meminimalkan gangguan tersebut. *Monitoring* selain berfungsi sebagai pengawasan juga berfungsi untuk merekam apa yang terjadi pada sistem yang di monitor dalam bentuk data tabel maupun grafik yang ditampilkan dalam bentuk *display*.

Monitoring yang terdapat dalam sistem ini, yaitu dilakukan dengan menggunakan *software Visual Basic*. Keluaran dari ATmega32 (*output* dari sensor tekanan) yang terbaca akan tersimpan dan terekam pada PC berupa grafik dan *database*.

Sehingga dengan adanya hasil *monitoring* ini dapat dijadikan sebagai acuan atau pedoman jika ada sesuatu yang terjadi pada *input* listrik karena variabel kontrol ini sangat berpengaruh terhadap kerja *press machine* beton.



**Gambar 2.1** Diagram blok *monitoring*<sup>[23]</sup>

Pada **Gambar 2.1** penjelasannya adalah bahwa untuk *monitoring flow* pada *input* pertama yang dibutuhkan adalah sensor. Sensor berperan sebagai alat yang bersentuhan langsung dengan variabel yang diukur kemudian hasil dari sensor tersebut dikonsikan dan diproses oleh STM32F4 *Discovery* kemudian ditampilkan pada *display* di LCD yang telah terprogram tampilan *monitoring*

## 2.2 Karbon Dioksida

Kandungan karbon dioksida di udara segar bervariasi antara 0,03% (300ppm) sampai dengan 0,06% (600 ppm) bergantung pada lokasi. Paparan berkepanjangan terhadap konsentrasi karbon dioksida yang sedang dapat menyebabkan efek-efek merugikan pada metabolisme kalsium *fosforus* yang menyebabkan peningkatan endapan kalsium pada jaringan lunak. Karbon dioksida beracun kepada jantung dan menyebabkan menurunnya gaya *kontraktil*. Pada konsentrasi tiga persen berdasarkan volume di udara, ia bersifat narkotik ringan dan menyebabkan peningkatan tekanan darah dan denyut nadi dan menyebabkan penurunan daya dengar. Pada konsentrasi sekitar lima persen berdasarkan volume udara ia menyebabkan stimulasi pusat pernapasan, pusing-pusing, kebingungan, dan kesulitan pernapasan yang diikuti sakit kepala dan sesak napas. Pada konsentrasi delapan persen, ia menyebabkan sakit kepala, keringatan, penglihatan buram, *tremor*, dan kehilangan kesadaran setelah paparan selama lima sampai sepuluh menit.<sup>[4]</sup>

Oleh karena bahaya kesehatan yang diasosiasikan dengan paparan karbon dioksida, Administrasi Kesehatan dan Keselamatan Kerja Amerika Serikat menyatakan bahwa paparan rata-rata untuk orang dewasa yang sehat selama waktu kerja 8 jam sehari tidak boleh melebihi 5.000 ppm (0,5%). Batas aman maksimum untuk balita, anak-anak, orang tua, dan individu dengan masalah kesehatan kardiopulmonari (jantung dan paru-paru) secara signifikan lebih kecil. Untuk paparan dalam jangka waktu pendek (di bawah 10 menit), batasan dari Institut Nasional untuk Kesehatan dan Keamanan Kerja Amerika Serikat (*NIOSH*) adalah 30.000 ppm (3%). *NIOSH* juga menyatakan bahwa konsentrasi karbon dioksida yang melebihi 4% adalah langsung berbahaya bagi keselamatan jiwa dan kesehatan<sup>[5]</sup>. Dalam ruangan tertutup yang dipenuhi orang, konsentrasi karbondioksida akan mencapai tingkat yang lebih tinggi daripada konsentrasi di udara bebas. Konsentrasi yang lebih besar dari 1.000 ppm akan menyebabkan ketidaknyamanan terhadap 20% penghuni dan ketidaknyamanan ini akan meningkat seiring dengan meningkatnya konsentrasi *CO2*.

Ketidaknyamanan ini diakibatkan oleh gas-gas yang dikeluarkan sewaktu pernapasan dan keringat manusia. Pada konsentrasi 2.000 ppm mayoritas penghuni akan merasakan ketidaknyamanan yang signifikan dan banyak yang akan mualmual dan sakit kepala.

**Tabel 2.1** Pengaruh Gas CO<sub>2</sub> terhadap Kesehatan<sup>[6]</sup>

KONSENTRASI (PPM)	PENGARUH TERHADAP KESEHATAN
350 – 450	Konsentrasi normal di luar ruangan
< 600	Konsentrasi yang masih ditolerir
600 – 1000	Menimbulkan kekakuan
1000 – 2500	Menimbulkan rasa kantuk
2500 – 5000	Mulai mengganggu kesehatan
5000	Konsentrasi maksimal yang diperbolehkan dalam waktu 8 jam
30000	Sedikit memabukkan, pernapasan dan denyut nadi bertambah, mual
50000	Sakit kepala dan gangguan penglihatan
100000	Tidak sadarkan diri, bahkan meninggal

### 2.3 Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini biasanya digunakan untuk memutar *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, *fan* angin) dan di industri. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus

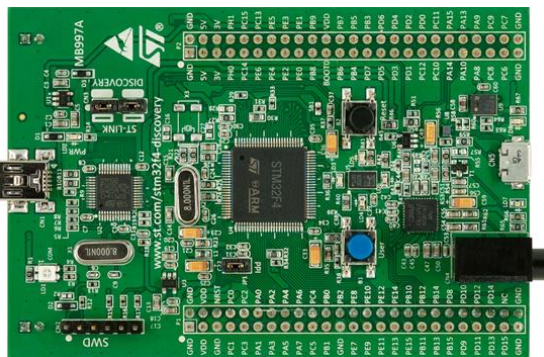
yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Pada tugas akhir ini, motor DC yang digunakan termasuk pada jenis *fan*, yaitu *DC Exhaust Fan* yang digunakan sebagai penyedot dan pembuang udara pada saat pengujian<sup>[7]</sup>.



**Gambar 2.2** DC Exhaust Fan

## 2.4 STM32F4 Discovery

STM32F4 Discovery merupakan suatu *development board* dengan mikrokontroller ARM Cortex M4 yang memiliki kecepatan sampai 168 MHz. Mikrokontroller ini memiliki 1 MByte Flash PEROM, 192 Kbyte SRAM. Gambar STM32F4 Discovery ditunjukkan oleh gambar 2.3.



**Gambar 2.3** STM32F4 Discovery

Pada modul STM32F4Discovery sudah dilengkapi ST-LINK/V2 programmer sehingga kita tidak perlu membeli modul programmer secara terpisah. ST-LINK/V2 pada modul ini mempunyai output SWD yang dapat digunakan untuk melakukan pemrograman modul external, konfigurasi dilakukan dengan merubah jumper pada board. Selanjutnya mari kita mengenal lebih dalam STM32F4Discovery dengan melihat feature-feature menarik apa saja yang terdapat pada modul ini<sup>[8]</sup>.

Fitur utama :

1. Mikrokontroler STM32F407VGT6 menampilkan 32-bit ARM® Cortex® -M4 dengan FPU inti, 1-Mbyte memori Flash, RAM 192-Kbyte dalam paket LQFP100
2. On-board ST-LINK / V2 pada STM32F4DISCOVERY (referensi tua) atau ST-LINK / V2-A pada STM32F407G-DISC1 (kode orde baru) USB ST-LINK dengan kemampuan re-pencacahan dan tiga antarmuka yang berbeda yakni *debug port*, *virtual port Com* (dengan kode orde baru saja), dan *mass storage* (dengan kode orde baru saja)
3. Dewan *power supply*: melalui bus USB atau dari tegangan suplai 5 Volt eksternal
4. Eksternal *power supply* aplikasi: 3 V dan 5 V
5. LIS302DL atau LIS3DSH ST MEMS *accelerometer 3-axis*
6. MP45DT02 ST-MEMS sensor audio yang *omni-directional microphone digital*
7. DAC audio yang CS43L22 dengan kelas yang terintegrasi sopir D speaker
8. Delapan LED, yakni:
  - LD1 (merah / hijau) untuk komunikasi USB
  - LD2 (merah) untuk 3,3 V daya pada
  - Empat LED pengguna, LD3 (oranye), LD4 (hijau), LD5 (merah) dan LD6 (biru)
  - 2 USB OTG LED LD7 (hijau) VBUS dan LD8 (merah) over-saat ini
9. Dua push-tombol (pengguna dan reset)

10. USB OTG FS dengan konektor *micro-AB extension header* untuk semua LQFP100 I / Os untuk koneksi cepat ke papan *prototyping* dan mudah menyelidik perangkat lunak bebas yang komprehensif termasuk berbagai contoh, bagian dari STM32CubeF4 paket atau STSW-STM32068 menggunakan standar perpustakaan warisan.

## 2.5 Sensor MG 811

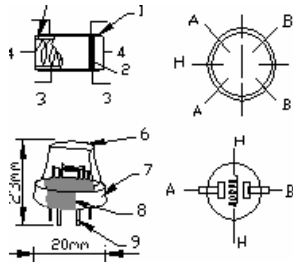
MG-811 merupakan sensor yang sensitif untuk mendeteksi gas CO<sub>2</sub> (Karbon Dioksida) namun kurang sensitif terhadap gas CO (karbon Monoksida). Sensor ini bisa diaplikasikan untuk mengontrol kualitas udara di dalam maupun di luar ruangan. Bentuk fisik dari sensor dapat dilihat pada Gambar 2.4. Sensor MG-811 memiliki 6 kaki pin yang terdiri dari 2 pin A, 2 pin B, dan 2 pin pemanas (Gambar 2.5.).

Sensor MG-811 ini di dalam nya terdapat komponen lilian pemanas yang berada dalam pipa keramik Al<sub>2</sub>O<sub>3</sub>. Struktur sensor terdiri dari bagian elektrolit padat dan pemanas (Gambar 2.6). Bagian elektrolit terbuat dari kation (Na<sup>+</sup>) yang berada diantara dua elektroda yang tersusun di atas pemanas. Elemen tersebut berfungsi sebagai pendeteksi gas CO<sub>2</sub><sup>[9]</sup>.

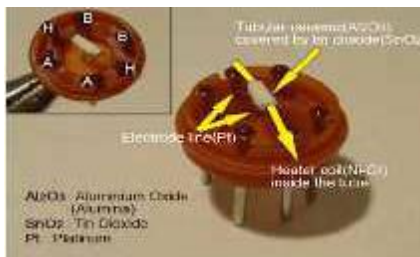


**Gambar 2.4** Bentuk Fisik MG-811





**Gambar 2.5** Struktur Sensor MG-811



**Gambar 2.6** Bagian Dalam Sensor MG-811

**Tabel 2.2** Spesifikasi Sensor MG 811

Symbol	Parameter Name	Technical	Remarks
$V_H$	Heating Voltage	$6.0 \pm 0.1 \text{ V}$	AC or DC
$R_H$	Heating Resistor	$30.0 \pm 5\% \Omega$	Room Temperature
$I_H$	Heating Current	@200mA	
$P_H$	Heating Power	@1200mW	
Tao	Operating Temperature	- $20 \dots +50^\circ\text{C}$	
Tas	Storage Temperature	- $20 \dots +70^\circ\text{C}$	
$E_{MF}$	Output	30—50mV	350-10000 ppm CO <sub>2</sub>

## 2.6 LCD 16x4

LCD (*Liquid Crystal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. Lapisan dari LCD terbuat dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan. Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD dan dilengkapi dengan memori dan *register* <sup>[10]</sup>.



**Gambar 2.7** LCD 16x4

**Tabel 2.3** Pin Konfigurasi LCD 16x4

<i>Pin No</i>	<i>Pin Name</i>	<i>Descriptions</i>
1	VSS	<i>Ground, 0V</i>
2	VDD	<i>Logic Power Supply</i>
3	V0	<i>Operating Voltage for LCD</i>
4	RS	<i>Data/Instruction Register</i>
5	R/W	<i>Read/Write</i>
6	E	<i>Enable Signal</i>
7	DB0	<i>Data Bit 0</i>
8	DB1	<i>Data Bit 1</i>

**Tabel 2.3** Pin Konfigurasi LCD 16x4 (lanjutan)

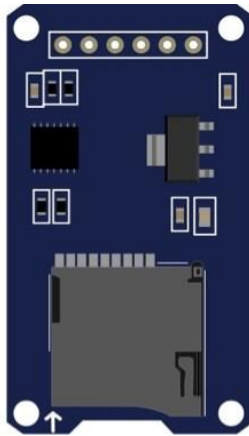
9	DB2	<i>Data Bit 2</i>
10	DB3	<i>Data Bit 3</i>
11	DB4	<i>Data Bit 4</i>
12	DB5	<i>Data Bit 5</i>
13	DB6	<i>Data Bit 6</i>
14	DB7	<i>Data Bit 7</i>
15	LED_A	<i>Anode</i>
16	LED_B	<i>Katode</i>

## 2.7 Micro SD Shield Module

*Micro Sd shield module* adalah kartu memori yang pada umumnya berukuran 11 x 15mm, dengan berbagai ukuran kapasitas yang digunakan untuk keperluan penyimpanan data maupun pembacaan data yang sudah ada didalamnya. Data tersebut bersifat digital yang dapat berupa data gambar, dokumen, video, maupun audio. Peringkat kecepatan transfer rate yang di kenal dengan *Speed Class* yang merupakan standar kecepatan yang ada pada SD Card. Untuk saat ini terdapat beberapa *speed class* antara lain :

- *Class 2* : dengan kecepatan 2 MB/s
- *Class 4* : dengan kecepatan 4 MB/s
- *Class 6* : dengan kecepatan 6 MB/s
- *Class 10* : dengan kecepatan 10 MB/s
- UHS 1 : dengan kecepatan 10 MB/s
- UHS 3 : dengan kecepatan 30 MB/s

Modul micro sd merupakan modul untuk mengakses memori card yang bertipe micro SD untuk pembacaan maupun penulisan data dengan menggunakan sistem antarmuka SPI (*Serial Parallel Interface*). Modul ini cocok untuk berbagai aplikasi yang membutuhkan media penyimpan data, seperti sistem absensi, sistem antrian, maupun sistem aplikasi *data logging* lainnya <sup>[11]</sup>.

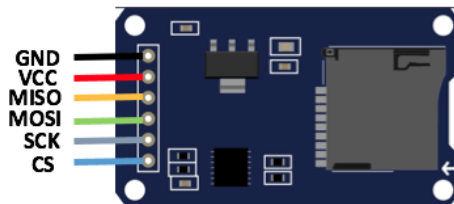


**Gambar 2.8** modul Micro Sd(1)

Fitur dan spesifikasi :

1. Mendukung pembacaan kartu memori SD Card biasa ( $\leq 2\text{G}$ ) maupun *SDHC card (high-speed card)* ( $\leq 32\text{G}$ )
2. Tegangan operasional dapat menggunakan tegangan 5V atau 3.3V
3. Arus operasional yang digunakan yaitu 80mA (0.2~200mA)
4. Menggunakan antarmuka SPI
5. Pada modul ini sudah terdapat 4 lubang baut guna untuk pemasangan pada rangkaian lainnya
6. Ukuran modul yaitu 42 x 24 x 12 mm

Control Interface



**Gambar 2.9** modul Micro Sd(2)

- GND : *negatif power supply*
- VCC : *positif power supply*
- MISO, MOSI, SCK : ***SPI bus***
- CS : *chip select signal pin*

## 2.8 RTC DS3231

Modul RTC (*Real Time Clock*) ini memiliki akurasi dan presisi yang sangat tinggi dalam mencacah waktu dengan menggunakan IC RTC DS3231 *extremely accurate temperature compensated* RTC (TCXO). DS3231 memiliki kristal internal dan rangkaian kapasitor *tuning* di mana suhu dari kristal dimonitor secara berkesinambungan dan kapasitor disetel secara otomatis untuk menjaga kestabilan detak frekuensi.

Pencacahan waktu pada solusi RTC lain dapat bergeser (*drift*) hingga hitungan menit per bulannya, terutama pada kondisi perubahan suhu yang ekstrim. Modul ini paling jauh hanya bergeser kurang dari 1 menit per tahunnya, dengan demikian modul ini cocok untuk aplikasi kritis yang sensitif terhadap akurasi waktu yang tidak perlu disinkronisasikan secara teratur terhadap jam eksternal <sup>[12]</sup>.



**Gambar 2.10** RTC (*Real Time Clock*)

## 2.9 *Modem GSM Wavecom Fastrack M1306B*

Wavecom adalah pabrikan asal Perancis yang bermarkas di kota Issy-lesMoulineaux, Perancis yaitu Wavecom.SA yang berdiri sejak 1993 bermula sebagai biro konsultan teknologi dan sistim jaringan nirkabel GSM, dan pada 1996 Wavecom mulai membuat desain daripada modul wireless GSM pertamanya dan diresmikan pada 1997, bentuk modul GSM pertama berbasis GSM dan pengkodean khusus yang disebut *ATcommand*. Modem Wavecom Fastrack ini cukup dikenal di Indonesia pada industri rumahan sampai sekala besar, mulai dari fungsi untuk SMS (Short Message Service) massal hingga penggerak perangkat elektronik, didukung pula dengan modem *wavecom* yang berjalan dengan baik di Quik Gateway pada software QuickSMS, kecepatan kirim 2-4 detik per sms. Beberapa fungsi kegunaan modem di masyarakat antara lain<sup>[22]</sup>:

1. *SMS Broadcast application*
2. *SMS Quiz application*
3. *SMS Polling*
4. *SMS auto-reply*
5. *M2M integration*
6. Aplikasi Server Pulsa
7. Telemetry
8. *Payment Point Data*
9. PPOB

*Wavecom M1306B* adalah GSM/GPRS modem yang siap digunakan sebagai modem untuk suara, data, fax dan SMS. Kelas ini juga mendukung 10 tingkat kecepatan transfer data. *Wavecom M1306B TCP/IP* dengan mudah dikendalikan dengan menggunakan perintah AT untuk semua jenis operasi karena mendukung fasilitas koneksi RS232 dan juga fasilitas *TCP/IP stacked*. Dapat dengan cepat terhubung ke port serial komputer *desktop* atau *notebook*. casing logam Wavecom M1306B TCP/IP menjadi solusi yang tepat untuk aplikasi berat seperti telemetri atau *Wireless Local Loop* (PLN metering & Telepon Umum). Ukurannya sangat kecil memudahkan dalam peletakkan di berbagai macam area, indoor/outdoor. Cocok sekali digunakan

pada aplikasi: Server Pulsa yang menghendaki kemampuan optimal dan usia pakai panjang, telemetri, *SMS gateway/broadcast* yang handal, PPOB PLN, ATM, *Payment Point Systems*, Metering Listrik. Modem GSM Serial *Wavecom Fastrack M1306B* memiliki dua type konektor yaitu serial dan USB<sup>[13]</sup>.

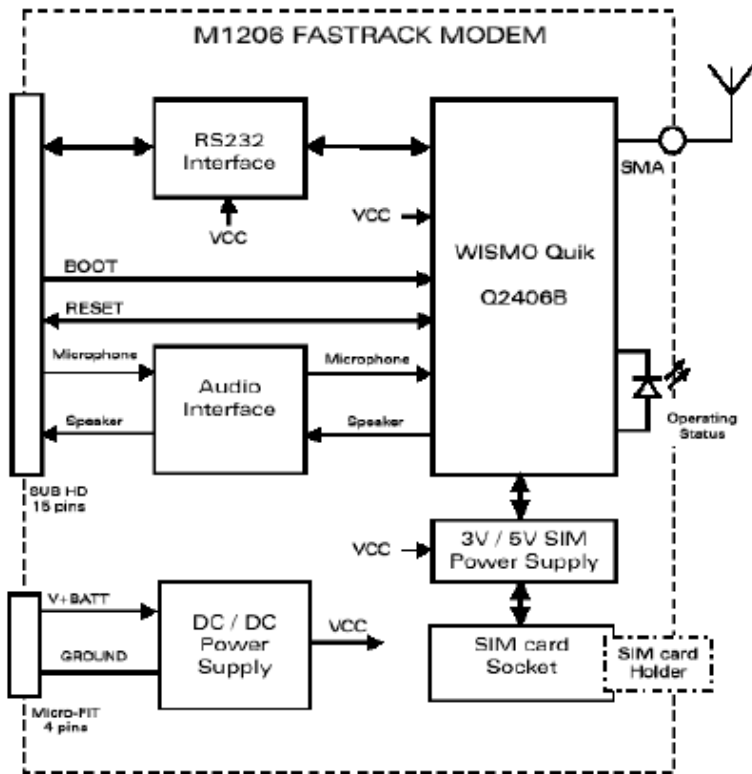


**Gambar 2.11** *Modem Wavecom Fastrack M1306B Serial*

Spesifikasi modem *WAVECOM FASTRACK M1206B*:

- *Dual-band GSM 900/1800MHZ & GPRS Class 10*
- *GSM Dual Band antenna*
- *Power Supply with 4 pin connector (untuk serial)*
- *Standard USB 2.0 interface (untuk USB)*
- *Input Voltage : 5V-32V*
- *Maximum transmitting speed 253KBps*
- *Support AT-Command*
- *Dimensi : 74×54×25mm*

Berikut ini arsitektur dari *Modem GSM Wavecom Fastrack M1306B*:



**Gambar 2.12** Arsitektur Modem Wavecom Fastrack M1306B Serial

### 2.10 MCB (*Miniature Circuit Breaker*)

MCB (*Miniature Circuit Breaker*) adalah saklar atau perangkat elektromekanis yang berfungsi sebagai pelindung rangkaian instalasi listrik dari arus lebih (*over current*). Terjadinya arus lebih ini, mungkin disebabkan oleh beberapa gejala, seperti: hubung singkat (*short circuit*) dan beban lebih (*overload*). MCB sebenarnya memiliki fungsi yang sama dengan sekering (*fuse*), yaitu akan memutus aliran arus listrik circuit ketika terjadi gangguan arus lebih. Yang membedakan keduanya adalah saat terjadi gangguan, MCB akan trip dan ketika rangkaian sudah normal,



MCB bisa di ON-kan lagi (reset) secara manual, sedangkan fuse akan terputus dan tidak bisa digunakan lagi.<sup>[14]</sup>



**Gambar 2.13** MCB Shukaku 220 V

### **2.11 ChibiOs**

ChibiOs adalah real-time OS yang digunakan untuk embedded system dan mendukung banyak mikroprosesor termasuk stm32. Dengan ChibiOs, kita bisa membangun program-program untuk akuisisi data, control system ataupun pemrosesan sinyal. Selain chibiOs, modul SYM32 ini ‘mencuri’ dari perangkat lunak open source lainnya seperti stm32flash, compiler gcc-arm-mone-eabi dan masih banyak lagi.<sup>[15]</sup>



**Gambar 2.14** Logo ChibiOS

ChibiOS juga mengintegrasikan komponen *Open Source* eksternal untuk menawarkan solusi lengkap untuk perangkat *embedded*.

Komponen ChibiOS tersedia di bawah lisensi *Open Source*, GPL3 atau *Apache* 2.0, ada juga beberapa pilihan lisensi komersial.  
[16]

## 2.12 Karakteristik Statik

Karakteristik statik yaitu karakteristik dari suatu instrument alat ukur yang tidak bergantung waktu. Karakteristik statik tersebut terdiri dari :<sup>[23]</sup>

### a. Akurasi

$$Akurasi = \left\{ 1 - Rata - Rata \frac{(pemb.std - pemb.alat)}{pembacaan std} \right\} \quad 2.3$$

Akurasi merupakan tingkat ketelitian suatu alat dalam memberikan hasil pengukuran.

### b. Presisi

Presisi merupakan kemampuan alat ukur untuk menampilkan nilai output yang sama pada pengukuran berulang.

$$Presisi = \left\{ 1 - Rata - Rata \frac{(pemb.alat - pemb.std)}{pembacaan alat} \right\} \quad 2.4$$

### c. Error

Selisis nilai pengukuran alat dengan nilai standar.

$$Error = pembacaan alat - pembacaan standar \quad 2.5$$

### d. Linearitas

Linearitas pada sensor merupakan perbandingan perubahan *output* terhadap perubahan *input* secara kontinyu. Untuk mendapatkan nilai linearitas dapat dihitung dengan persamaan berikut.

$$K = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \quad 2.6$$

$$a = O_{min} - (K \cdot I_{min}) \quad 2.7$$

**e. Hysterisis**

*Hysterisis* merupakan .

$$\%H = \frac{\hat{H}}{O_{max} - O_{min}} \times 100\% \quad 2.8$$

**f. Sensitivitas**

Sensitivitas merupakan penunjukan seberapa jauh kepekaan sensor terhadap kuantitas yang diukur. Sensitivitas sering juga dinyatakan sebagai bilangan yang menunjukkan perubahan keluaran (*output*) terhadap perubahan masukan (*input*).

$$Sensitivitas (K) = \frac{\Delta o}{\Delta I} \quad 2.9$$

## 2.13 Teori Ketidakpastian

Sumber-sumber ketidakpastian yang turut memberikan kontribusi selain ada pada diri manusia sendiri sebagai pelaku pengukuran/kalibrasi juga pada alat-alat bantu (kalibrator) yang digunakan, juga resolusi alatnya serta pengaruh suhu lingkungan.

[23]

**a. Koreksi**

Koreksi dapat diperoleh dengan persamaan berikut :

$$\text{Koreksi} = \text{Pembacaan standard} - \text{Pembacaan alat} \quad 2.10$$

**b. Standard deviasi**

$$\sigma = \frac{\sqrt{\Sigma(Di - D')^2}}{n-1} \quad 2.11$$

dimana :

Di = koreksi alat ukur

Di' = rata-rata koreksi

n = Banyak range pengukuran

**c. Analisa Type A, ( $U_a$ )**

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran. Adapun persamaannya adalah sebagai berikut:

$$U_{a1} = \frac{\sigma_{maks}}{\sqrt{n}} \quad 2.12$$

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} \quad 2.13$$

**d. Analisa Type B, ( $U_b$ )**

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur. Adapun persamaannya adalah sebagai berikut:

$$U_{b1} = \frac{Resolusi/2}{\sqrt{3}} \quad 2.14$$

Dimana :

$SSR = \text{Sum Square Residual}$

$$U_{b2} = \frac{a}{k} \quad 2.15$$

**e. Ketidakpastian Kombinasi ( $U_C$ )**

$U_C$  merupakan Ketidakpastian kombinasi dari ketidakpastian tipe A dan ketidakpastian tipe B. Adapun persamaan dari ketidakpastian kombinasi adalah:  $U_C =$

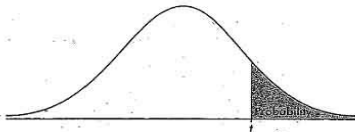
$$\sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \quad 2.16$$

**f. Ketidakpastian Diperluas**

Hasil akhir kalibrasi adalah ketidakpastian diperluas sehingga alat ukur tersebut dapat diketahui ketidakpastiannya melalui  $U_{expand}$ . Persamaan  $U_{expand}$  adalah:

$$U_{expand} = k \cdot U_C \quad 2.17$$

Untuk mencari nilai  $k$ , maka melihat table t student sesuai dengan *confidence level* 95%. Tabel T *student* dapat dilihat pada gambar 2.7.

TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

df	Tail probability <i>p</i>									
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937
60	.679	.848	1.045	1.295	1.671	2.000	2.099	2.390	2.660	2.915
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%
										99.8%
										99.9%

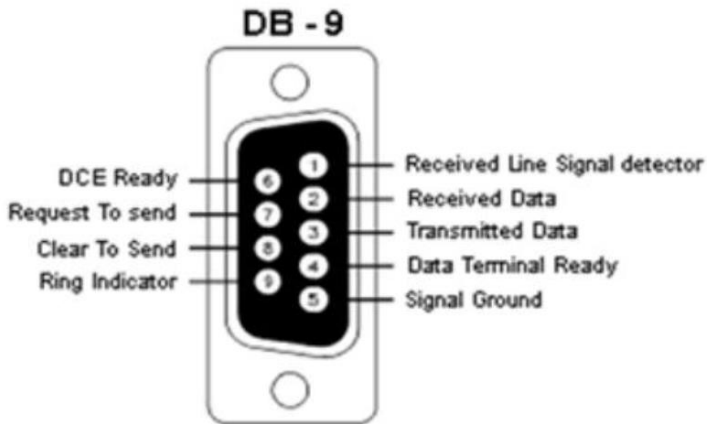
Confidence level *C*Gambar 2.15 Tabel *T-student*g. **V efektif**

$$V_{\text{eff}} = \frac{(U_c)^4}{\frac{\sum (U_i)^4}{V_i}}$$

2.18

## 2.14 Konfigurasi Port Serial

Konektor DB-9 pada bagian belakang komputer adalah port serial RS232 yang biasa dinamai dengan COM1 dan COM2. <sup>[18]</sup>



**Gambar 2.16** Port Serial

**Tabel 2.4** Konfigurasi Port Serial

Pin	Nama Sinyal	Direction	Keterangan
1	DCD	In	<i>Data Carrier Detect/Receive Line Signal Detect</i>
2	RxD	In	<i>Receive Data</i>
3	TxD	Out	<i>Transmit Data</i>
4	DTR	Out	<i>Data Terminal Ready</i>
5	GND	-	<i>Ground</i>
6	DSR	In	<i>Data Set Ready</i>
7	RTS	Out	<i>Request to Send</i>
8	CTS	In	<i>Clear to Send</i>
9	RI	In	<i>Ring Indicator</i>

Berikut ini keterangan mengenai fungsi saluran RS232 pada konektor DB-9:

1. *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk.
2. *Receive Data*, digunakan DTE untuk menerima data dari DCE.
3. *Transmit Data*, digunakan DTE untuk mengirimkan data ke DCE.
4. *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya.
5. *Signal Ground*, saluran ground
6. *DCE ready* adalah sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.
7. *Request to Send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
8. *Clear to Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data.
9. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa sebuah stasiun menghendaki hubungan dengannya.

### **2.15 Qt Creator**

*QT Creator* merupakan *cross-platdorm IDE (Integrated Development Environment )* yang lengkap untuk pengembangan aplikasi dengan target berbagai platform desktop dan berbagai *platform mobile*. *QT Creator* dapat diinstall pada Linux, OS X dan *Micorosoft Windows*.



**Gambar 2.17** Logo *QT Creator*

*QT Creator* merupakan IDE yang menyediakan *tools* untuk mendisain dan mengembangkan aplikasi menggunakan *framework* aplikasi Qt. QT merancang tampilan dan mengembangkan aplikasi sekali kemudian menyebarkan aplikasi tersebut ke berbagai *platform desktop* dan *platform mobile*. *QT Creator* menyediakan *tool-tool* menyeluruh dalam mengembangkan aplikasi dimulai dari memulai *project* dan menyebarkan aplikasi ke berbagai target *platform*.

Salah satu kelebihan *QT Creator* adalah memungkinkan sebuah team pengembang aplikasi bekerja sama mengembangkan aplikasi dari berbagai *platform* dengan menggunakan *tool-tool* dan *debugging* yang sama.

Saat memulai *project*,

1. Pertama-tama kita harus memilih user *interface* apakah menggunakan *QT Quick* atau *QT Widgets*.
2. Kedua kita dapat memilih bahasa untuk dipakai dalam logika pemrograman. Kita dapat memilih C++ atau *Javascript*.

Pada waktu kita memulai *project* maka ada *wizard* yang mengarahkan langkah-langkah dalam proses pembuatan *project*. *Wizard* akan mengarahkan setting yang diperlukan oleh *project* tersebut dan membuat file-file yang kita butuhkan. Tentu saja kita dapat melakukan kustomisasi mengenai file-file yang telah dihasilkan.



Sebagian besar *wizard* membuat *project* yang menggunakan *Qt build tool* yaitu *qmake*. Program *qmake* merupakan *cross-platform system* untuk melakukan otomatisasi yang menyederhanakan pengembangan aplikasi dengan target berbagai *platform*<sup>[19]</sup>..

### 2.16 *MicroSD*

*MicroSD* merupakan salah satu jenis kartu memori eksternal yang memiliki ukuran paling kecil. *MicroSD* ini dikembangkan oleh *SD card assosiation* yang fungsinya sebagai media penyimpanan data *portable device*. Varian kapasitas *microSD* mulai dari kapasitas 1 GB sampai 128 GB. Biasanya besar kecilnya kapasitas *microSD* sangat mempengaruhi harga *microSD*.



**Gambar 2.18** Logo *microSD*

Begitu pentingnya fungsi *microSD* dalam menunjang penyimpanan data pada *smartphone* dan *android*. Bahkan seiring berkembangnya *smartphone* dan *android* sehingga kehadiran *microSD* semakin merebak. Maka dari itu kemampuan penyimpanan data sangat diperhitungkan. Dimana *microSD* juga memiliki class memori untuk membedakan kecepatan dan keunggulan masing masing *microSD*<sup>[20]</sup>.

### 2.17 *Notepad++*

*Notepad++* adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. *Notepad++* menggunakan komponen *Scintilla*

untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi *Microsoft Windows*.



**Gambar 2.19** Logo *Notepad++*

Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, *Notepad++* juga dilisensikan sebagai perangkat *free*. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena *sourceforge.net* sebagai layanan yang memfasilitasi *Notepad++* membebaskannya untuk digunakan.

Beberapa daftar bahasa program yang didukung oleh *Notepad++* adalah C, C++, Java, C#, XML, HTML, PHP, *Javascript*. Sebenarnya masih banyak lagi bahasa program yang didukung, namun penulis baru mencoba *Notepad++* dengan bahasa program yang diatas <sup>[21]</sup>.

Fitur-fitur:

1. WYSIWYG
2. *User Defined Syntax Highlighting*
3. *Multi-Document Tabs*
4. *Regular Expression Search/Replace supported*
5. *Full Drag N' Drop supported*
6. *Dynamic position of Views*

7. File Status Auto-detection
8. *Zoom in and zoom out*
9. *Multi-Language environment supported*
10. *Bookmark*
11. *Brace and Indent guideline Highlighting*
12. *Macro recording and playback*

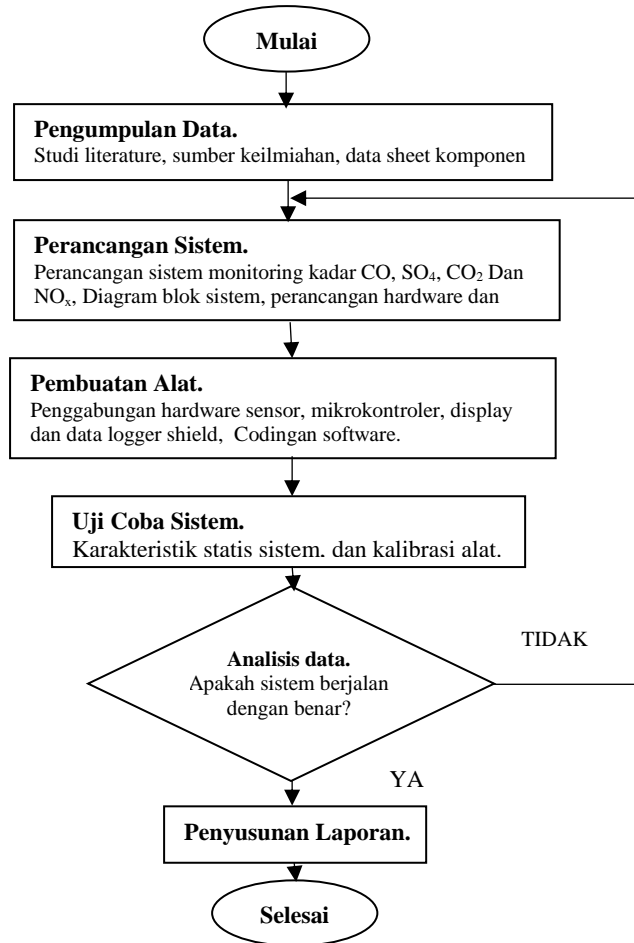
**(Halaman ini sengaja dikosongkan)**

### BAB III

## PERANCANGAN DAN PEMBUATAN ALAT

### 3.1 *Flowchart* dan Diagram Blok Perancangan Alat

Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* yang dapat dilihat pada gambar 3.1 di bawah ini.

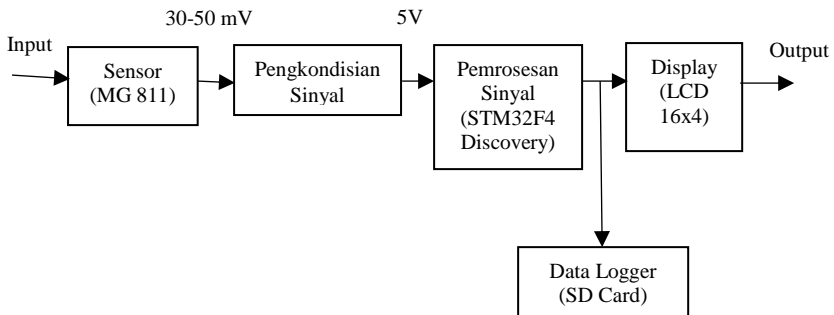


**Gambar 3.1** *Flowchart* Pengerjaan Tugas Akhir

*Flowchart* diatas merupakan *flowchart* pengerjaan tugas akhir mulai dari *start* hingga selesai. Tahap awal pada *flowchart* tugas akhir ini dimulai dengan adanya studi literatur sebagai upaya pemahaman terhadap materi yang menunjang tugas akhir mengenai “Rancang Bangun Sistem Monitoring Konsentrasi Gas Karbon Dioksida ( $\text{CO}_2$ ) Sebagai Emisi Gas Buang Menggunakan Sensor MG 811 Berbasis Mikrokontroler STM32F4 Discovery”. Setelah melakukan studi literatur, selanjutnya adalah melakukan perancangan sistem dan mempersiapkan komponen yang dibutuhkan. Kemudian dibuat perancangan *hardware*, *software*, dan mekanik dari sistem monitoring gas karbon dioksida ( $\text{CO}_2$ ) berbasis STM32F4 Discovery. Setelah itu dari sistem monitoring gas karbon dioksida ( $\text{CO}_2$ ) yang telah dibuat, dilakukan pengujian alat dengan memberi *input gas* pada box , sehingga dapat diketahui konsentrasi gas karbon dioksida ( $\text{CO}_2$ ). Apabila semua rancang bangun sistem monitoring gas karbon dioksida ( $\text{CO}_2$ ) dapat bekerja dengan baik, maka selanjutnya dilakukan pengambilan data pada *plant monitoring gas*. setelah pembuatan rancangan telah selesai dengan hasil yang sesuai dengan yang diinginkan, kemudian dilakukan analisis data dengan memanfaatkan hasil dari uji performansi dan sistem pengendalian. Setelah semua hasil yang diinginkan tercapai mulai dari studi literatur hingga analisa data dan kesimpulan dicantumkan dalam sebuah laporan.

Diagram merupakan pernyataan hubungan yang berurutan dari suatu atau lebih komponen yang memiliki kesatuan kerja tersendiri, dan setiap blok komponen mempengaruhi komponen lainnya. Diagram blok merupakan salah satu cara yang paling sederhana untuk menjelaskan cara kerja dari suatu sistem. Dengan diagram blok dapat menganalisa cara kerja rangkaian dan merancang *hardware* yang akan dibuat secara umum.

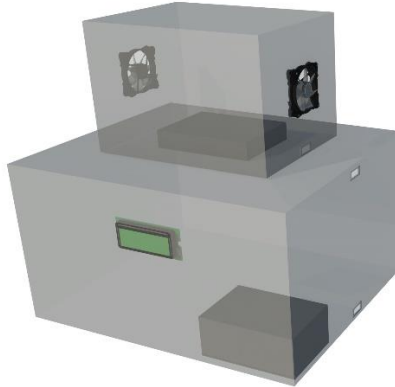
Adapun diagram blok dari sistem yang dirancang, seperti yang diperlihatkan pada gambar 3.2.



**Gambar 3.2** Diagram Blok Sistem monitoring gas buang

Gambar 3.2 di atas merupakan gambar diagram blok sistem monitoring emisi gas buang berupa konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sensor gas MG-811 yang dikontrol oleh mikrokontroler STM32F4 Discovery sebagai pengkondisian sinyal dan tempat pemrosesan sinyal, dan sensor MG 811 sebagai sensor untuk sensing gas. Ketika sensor MG 811 telah mendapat input gas maka oleh sensor akan diubah menjadi tegangan dan kemudian output ini dibaca oleh ADC (*Analog to Digital Converter*) internal dari mikrokontroler STM32F4 yang kemudian data dikalkulasikan dengan rumusan tertentu sehingga pada tahap berikutnya sistem dapat menentukan apakah tekanan yang dihasilkan sudah sesuai dengan nilai yang seharusnya atau belum. Konsentrasi gas yang terbaca oleh sensor akan ditampilkan ke display LCD (*Liquid Crystal Display*) 16x4 yang ditampilkan dalam satuan ppm. Dalam perancangan mekanik terdiri dari beberapa komponen mekanik seperti box, LCD dan sensor. Box disini berfungsi sebagai tempat komponen dan rangkaian elektrik. Selain itu terdapat juga perancangan elektrik yang meliputi rangkaian power supply, rangkaian minimum sistem mikrokontroler, rangkaian driver sensor dan rangkaian LCD. Dalam rancang bangun sistem monitoring konsentrasi gas karbon Dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang pada kendaraan bermotor dimana terdiri dari beberapa bagian penting yaitu sensor, sinyal pengkondisian, sinyal pemrosesan dan representasi data atau

*display* berupa LCD (*Liquid Crystal Display*) 16x4 yang ditampilkan dalam satuan ppm.



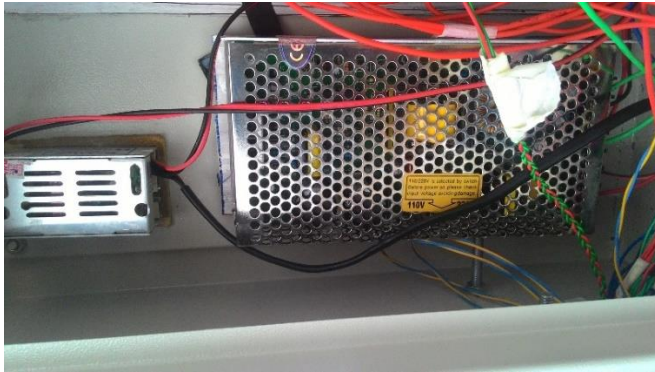
**Gambar 3.3** Skematik *Plant* monitoring gas

### 3.2 Rangkaian *Power Supply*

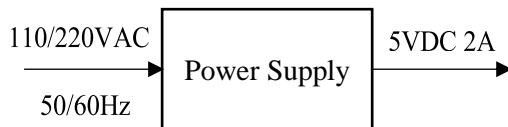
*Power supply* merupakan sumber tenaga yang dibutuhkan suatu rangkaian elektronika untuk bekerja. Besar *power supply* ini tergantung oleh spesifikasi dari alat masing-masing. Pada perancangan sistem pengendali ini *power supply* digunakan untuk men-*supply* rangkaian mikrokontroler STM32F4 discovery, rangkaian sensor MG 811, rangkaian sensor MQ 7, rangkaian sensor MQ 136, rangkaian sensor MQ 135, modul RTC DS3231, modul sd card shield, modul sms gateway, dan LCD 16x4.

Pada rangkaian *power supply* pada umumnya sering menggunakan IC regulator dalam mengontrol tegangan yang diinginkan. Regulator tegangan menjadi sangat penting gunanya apabila mengaplikasikan *power supply* tersebut untuk rangkaian-rangkaian yang membutuhkan tegangan yang sangat stabil. Misalkan untuk sistem *digital*, terutama untuk mikrokontroler yang sangat membutuhkan tegangan dan arus yang stabil.





**Gambar 3.4** *Power Supply DC*



**Gambar 3.5** Blok Diagram *Power Supply*

*Power supply* untuk tegangan DC digunakan sebagai *supply* untuk perangkat yang membutuhkan tegangan DC 12 V dan 5 V. IC regulator yang umum digunakan untuk mengontrol tegangan adalah IC keluarga 78XX. IC ini dapat mengontrol tegangan dengan baik. Keluaran tegangan yang diinginkan tinggal melihat tipe yang ada. Misalkan tipe 7805 dapat memberikan keluaran tegangan 5 V dengan toleransi  $\pm 1$  dengan arus keluaran maksimal 1500 mA.

### 3.3 Perancangan *Sensor MG 811*

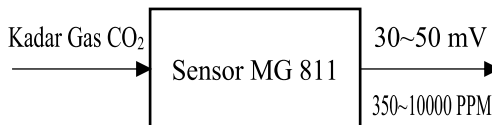
Sensor MG-811 digunakan untuk mendeteksi senyawa gas karbon dioksida ( $\text{CO}_2$ ). Penggunaan sensor MG-811 menggunakan prinsip reaksi elektrokimia untuk tegangan outputnya. Reaksi elektrokimia yang terjadi menghasilkan *emf* (*electromotive force*) diantara dua elektroda. Perubahan nilai *emf* menunjukkan perubahan

kosentrasi karbon dioksida( $\text{CO}_2$ ) yang dideteksi sensor dan untuk gambar rangkaian pengkondisian sinyal sensor MG-811 dapat dilihat. Elektroda A dan B dipakai untuk reaksi elektrokimia. Tegangan sebesar 5 volt DC diberikan ke pemanas sensor yang akan menghasilkan *emf* ketika elektrodamendeteksi adanya senyawa karbon dioksida( $\text{CO}_2$ ).



**Gambar 3.6** Karbon dioksida( $\text{CO}_2$ ) Gas Sensor Module

Sensor ini yang bereaksi terhadap kadar gas karbon dioksida( $\text{CO}_2$ ) yang terdapat dalam udara. *Sensor* ini nantinya akan mengirimkan data kepada mikrokontroler STM32F4 *Discovery* dan data hasil pembacaan konsentrasi gas tersebut dapat disimpan melalui *data looger* pada *SD Card* dan dikirimkan melalui *SMS Gateway* dan selanjutnya dapat ditampilkan dalam *Liquid Crystal Display* (LCD).



**Gambar 3.7** Blok Diagram *Sensor MG 811*

Sensor Gas MG 811 merupakan untuk *sensor* yang digunakan mengetahui kadar atau konsentrasi gas karbon dioksida( $\text{CO}_2$ ),

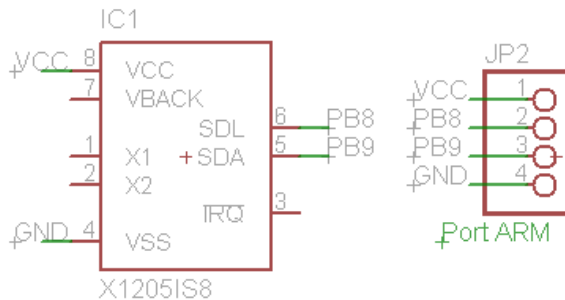
*sensor* ini bisa mengukur dengan *range* 350~10000ppm CO<sub>2</sub>, dan tegangan *output* berada 30~50 mV.

Pada modul sensor MG 811 ini terdapat 6 kaki. Pada kaki 1 merupakan Vin yang berfungsi sebagai tegangan sumber untuk menhidupkan sensor. Pada kaki 2 merupakan Aout yang berfungsi sebagai keluaran analog dari sensor. Pada kaki 3 merupakan Ground.

Dalam perancangannya, Modul sensor MG 811 dihubungkan dengan mikrokontroller STM32F4 Discovery yang sudah tersambung dengan shield yang telah dibuat. kaki Vin pada modul disambungkan ke terminal Vcc pada shield STM32F4 Discovery. kaki Ground pada modul MG 811 disambungkan ke terminal Ground STM32F4 Discovery.

### **3.4 Perancangan *Real Time Clock (RTC)***

RTC Module RTC digunakan untuk mempertahankan data hari dan tanggal supaya terus beroperasi walaupun catu daya dimatikan. RTC membantu arduino untuk menambahkan data hari dan tanggal untuk keperluan history alat. Jenis RTC yang paling banyak dipakai adalah tipe DS1307 dan tipe DS3231. Secara keseluruhan spesifikasi keduanya memiliki kesamaan, satu perbedaan yang membuat tipe DS3231 memiliki keunggulan adalah karena memiliki integrated temperature compensated crystal oscillator (TCXO) dan menggunakan crystal oscillator internal, sehingga perhitungan hari dan tanggalnya lebih akurat tidak dipengaruhi temperatur sekitar dibandingkan tipe DS1307 yang menggunakan crystal oscillator eksternal. Hal tersebut dapat menyebabkan perhitungan hari dan tanggal DS1307 akan bergeser 1 menit setiap tahunnya. Oleh karena itu RTC yang digunakan dalam tugas akhir ini adalah tipe DS3231.



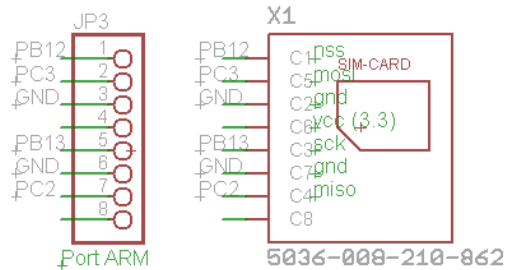
**Gambar 3.8** Skematik RTC pada ARM

IC RTC DS3231 sebagai input pemberi referensi waktu terhadap data yang akan diperoleh. Cara kerjanya adalah alamat dan data ditransmisikan secara serial melalui sebuah jalur data dua arah I2C. Karena menggunakan jalur data I2C maka hanya memerlukan dua buah pin saja untuk berkomunikasi yaitu pin data dan pin untuk sinyal clock (SDA dan SCL) sehingga STM32F4 discovery dapat mengolah data dan clock yang diterima dari RTC untuk dijadikan referensi waktu. Pin SDA dihubungkan ke port PB9 dan pin SCL ke port PB8 STM32F4 discovery.

### 3.5 Perancangan Data Logger

Data Logger Shield adalah perangkat yang memiliki fungsi sebagai media penghubung antara mikrokontroler dengan SD Card, dimana pada data logger shield terdapat SD Card slot dan RTC (Real Time Clock).

Perancangan ini digunakan dalam sistem monitoring ini. Modul SD card digunakan sebagai data logger dalam penyimpanan informasi dari hasil ppm yang telah diukur. Penyimpanan data ini disimpan dalam format excel atau .csv sehingga dapat mudah dilihat. Terdapat 6 pin dalam modul sd card ini yakni Ground, VCC, MISO, MOSI, SCK, dan CS.



**Gambar 3.9** Skematik SD card pada ARM

Modul micro sd merupakan modul untuk mengakses memori card yang bertipe micro SD untuk pembacaan maupun penulisan data dengan menggunakan sistem antarmuka SPI (Serial Parallel Interface). Modul ini cocok untuk berbagai aplikasi yang membutuhkan media penyimpanan data, seperti sistem absensi, sistem antrian, maupun sistem aplikasi data logging lainnya.

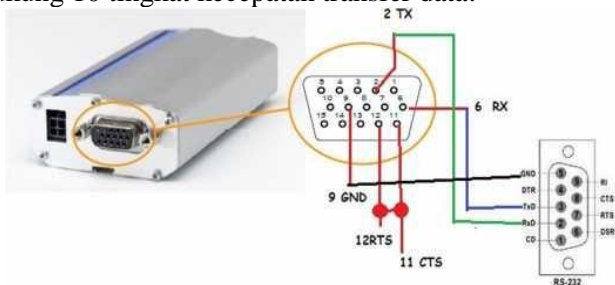
Dalam perancangannya, menghubungkan pin pada Modul SD Card kedalam mikrokontroller STM32F4 Discovery yang terhubung dengan shield yang telah dibuat. Pin Ground pada modul SD Card disambungkan ke Ground STM32F4 Discovery. Pin VCC pada modul SD Card disambungkan ke pin 5 volt STM32F4 Discovery. Pin MISO pada modul SD Card disambungkan ke pin PC2 STM32F4 Discovery. Pin MOSI pada modul SD Card disambungkan ke pin PC3 STM32F4 Discovery. Pin SCK pada modul SD Card disambungkan ke pin PB13 STM32F4 Discovery dan Pin CS pada modul SD Card disambungkan ke pin PB12 STM32F4 Discovery.

### 3.6 Perancangan Modem Wavecom

Prinsip kerja modem wavecom M1306B fastrack sama dengan modem GSM untuk mengirim pesan singkat (SMS) pada umumnya, yaitu pesan tidak langsung dikirim ke ponsel tujuan, akan tetapi dikirim terlebih dahulu ke SMS Center (SMSC) yang biasanya berada di kantor operator telepon, baru kemudian pesan tersebut diteruskan ke ponsel tujuan. Dengan adanya SMSC, dapat diketahui status pesan SMS yang telah dikirim, apakah telah

sampai atau gagal. Pada perancangan ini modem wavecom M1306B fastrack dihubungkan ke mikrokontroler sebagai pengganti komputer yang memberikan perintah untuk mengirimkan SMS. Mikrokontroler mengirim data (Isi SMS dan no.tujuan) ke modem wavecom M1306B melalui RS232, kemudian modem mengirim data tersebut ke SMS center yang akan menyampaikan ke no.ponsel tujuan.

Pada bagian rancang bangun monitoring gas ini terdapat modem wavecom M1306B serial RS232. Dimana modem ini banyak digunakan oleh pengguna layanan sms gateway untuk broadcast sms, kirim sms massal dan compatible dengan engine sms seperti gammu dan quick gateway. Modem ini juga dilengkapi dengan AT Command untuk semua jenis operasi karena mendukung fasilitas koneksi RS232 dan dapat dengan cepat terhubung ke port serial komputer maupun mikrokontroler sehingga sangat mudah untuk dikendalikan. Kelas ini juga mendukung 10 tingkat kecepatan transfer data.



**Gambar 3.10** Konfigurasi port modem wavecom

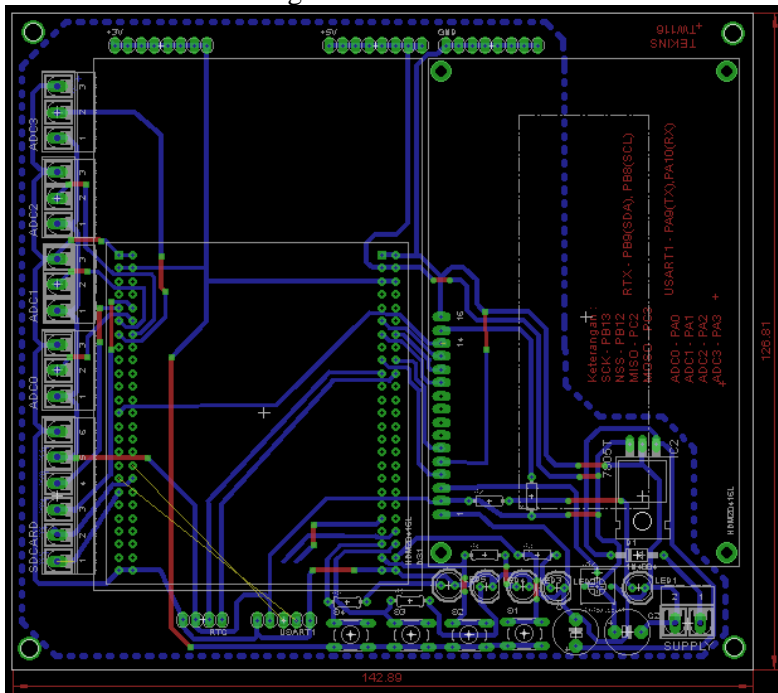
Karena tegangan keluaran dari modem wavecom tidak compatible dengan mikrokontroler maka harus dihubungkan dengan modul RS232 agar sesuai dengan tegangan pada mikrokontroler.

Dalam perancangan penggunaan modem wavecom M1306B serial RS232 dilakukan dengan cara mengambil pin Rx, Tx, dan Ground dari serial RS232 modem wavecom M1306B dan menghubungkan pin Rx, Tx, dan Ground ke dalam mikrokontroler

STM32F4 Discovery yang terhubung dengan shield yang telah dibuat. Pin Ground pada modul disambungkan ke Ground STM32F4 Discovery. Pin VCC pada modul SD Card disambungkan ke pin 5 volt STM32F4 Discovery. Pin Rx pada modul SD Card disambungkan ke pin PA10 STM32F4 Discovery. Pin Tx pada modul SD Card disambungkan ke pin PA9 STM32F4 Discovery .

### 3.7 PCB Designer

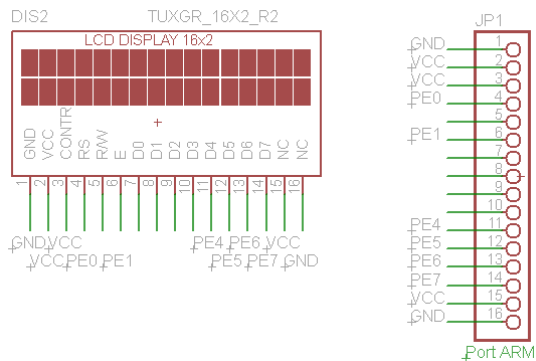
PCB designer adalah salah satu software yang berguna untuk membuat jalur PCB. Program ini termasuk ringan digunakan sehingga mudah dalam penggunaannya. Contoh tampilan dari software ini adalah sebagai berikut.



Gambar 3.11 Layout PCB Designer

### 3.8 Perancangan *Liquid Crystal Display (LCD)*

LCD yang digunakan dalam *training kit* yang penulis rancang ini menggunakan LCD tipe 16 x 4 (16 buah kolom yang tersusun berderet dalam 4 baris). LCD merupakan salah satu piranti yang penulis gunakan pada *training kit* yang penulis rancang sebagai salah satu media output, dimana data yang diset oleh user pada mikrokontroler dapat dilihat langsung pada LCD melalui port yang tersedia, pada rangkaian LCD yang penulis buat terdapat sebuah potensiometer pada kaki 3 LCD dan dioda pada kaki 15 LCD. Potensiometer berfungsi sebagai pengatur kontras LCD, sedangkan dioda berfungsi sebagai back light. Berikut skematik rangkaian Rangkaian LCD yang terdapat pada *training kit* Mikrokontroler yang penulis rancang ::



**Gambar 3.12** Rangkaian LCD pada STM32F4 Discovery

Gambar 3.5 diatas merupakan skematik rangkaian lcd dengan STM32F4 *Discovery*. *Display LCD* adalah suatu *modul* penampil. Dalam hal ini untuk menampilkan data yang terdeteksi pada *sensor* sehingga dapat ditampilkan data berupa *digital* yang menunjukkan *display* nilai kadar konsentrasi gas pada *plant monitoring* gas. LCD yang digunakan dalam *plant monitoring* gas ini adalah LCD 16 kolom x 4 baris.

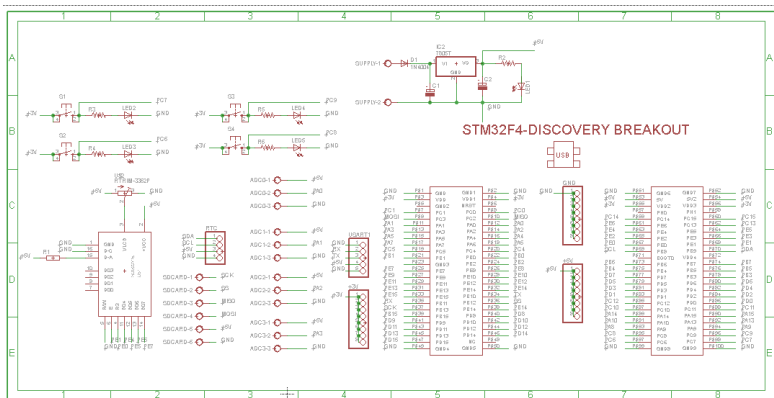




**Gambar 3.13** Penempatan LCD untuk *Display*.

### 3.9 Perancangan dan Pembuatan *Software*

Pada perancangan dan pembuatan *software* dilakukan dengan menggunakan *software eagle*, dengan mengintegrasikan *sensor* MQ 136, *SD Card*, *LCD 16x4*, *RTC*, *SMS Gateway* dan mikrokontroler *STM32F4 Discovery*. Berikut merupakan rangkaian skematik mikrokontroler *STM32F4*, *sensor* MG 811, *SD Card*, *RTC*, *Uart* (*SMS Gateway*) dan *LCD 16x4* menggunakan *software eagle*.

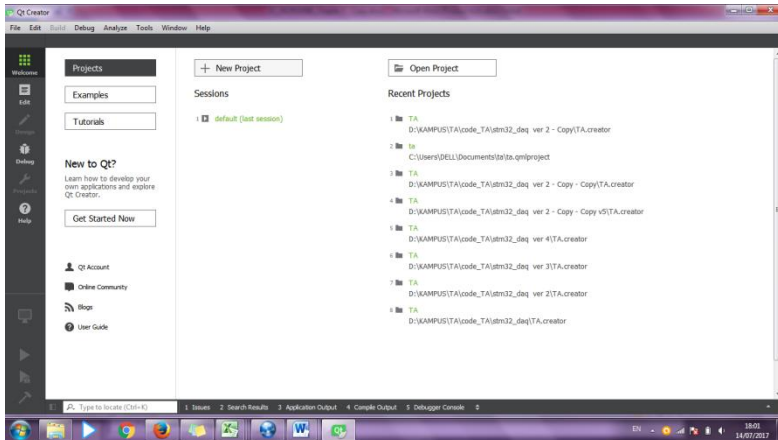


**Gambar 3.14** Rangkaian Skematik Mikrokontroler *STM32F4*, *sensor* MG 811, *SD Card*, *RTC*, *Uart* (*SMS Gateway*) dan *LCD 16x4*

Kemudian untuk pemrograman menggunakan *Qt Creator* dilakukan dengan langkah-langkah sebagai berikut:

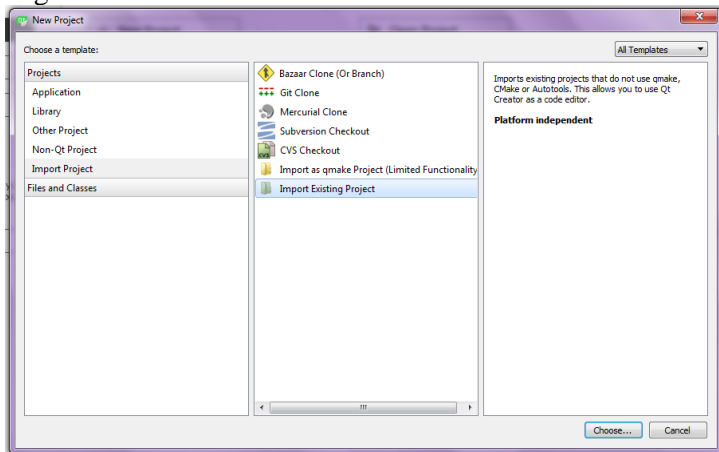
a. *Qt Creator* dibuka.

- b. Klik *New project*. Kemudian akan muncul gambar seperti dibawah ini.



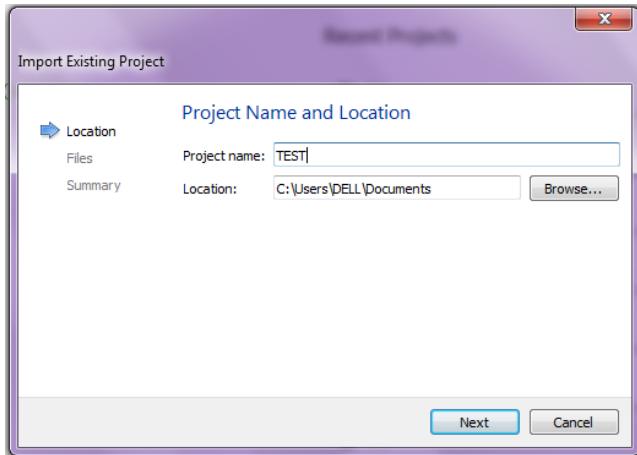
**Gambar 3.15** *Create New Project*

- c. Pilih *import project*. Lalu klik *import existing project* seperti gambar dibawah ini:



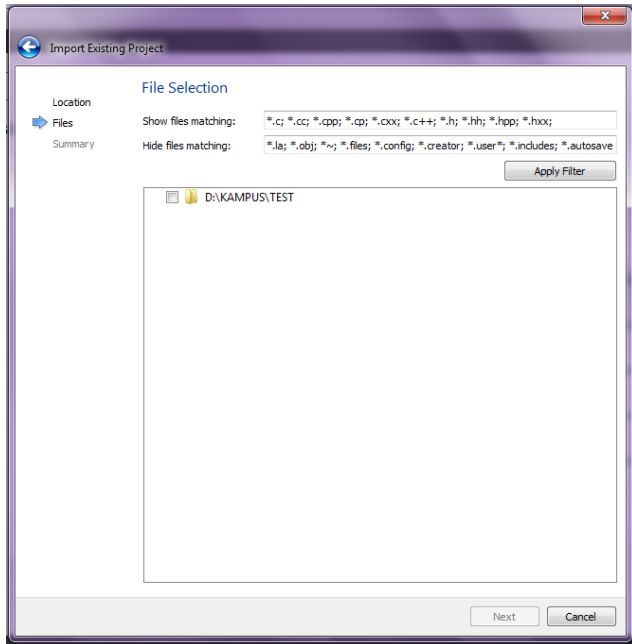
**Gambar 3.16** *Project Qt Creator*

- d. Kemudian akan muncul dialog *import existing project*. Beri nama *project* sesuai yang diinginkan dan pilih lokasi yang diinginkan dalam laptop. Kemudian klik *next*.



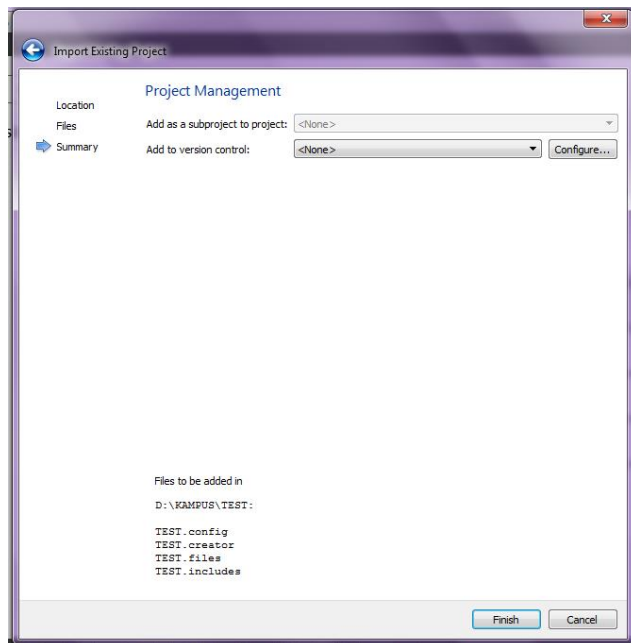
**Gambar 3.17** *import existing project name and location*

- e. Kemudian akan muncul kotak *file selection*. Setelah itu centang *location file* dan klik *next*.



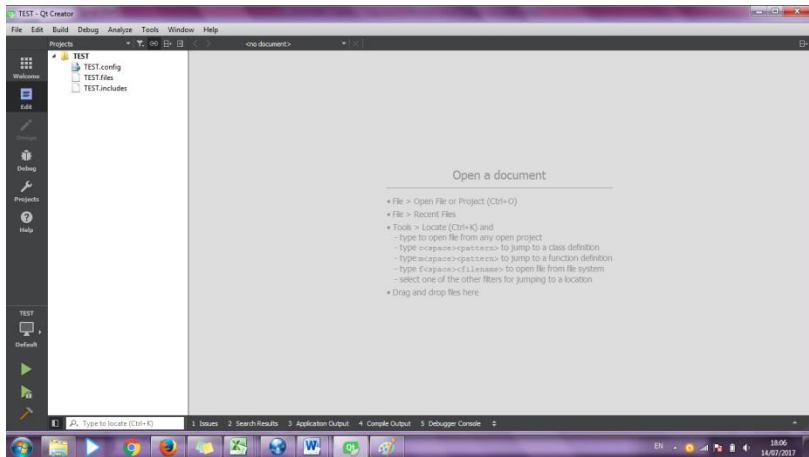
**Gambar 3.18** *import existing project file selection*

- f. Setelah itu akan muncul kotak *project management*. Lalu klik *finish*.



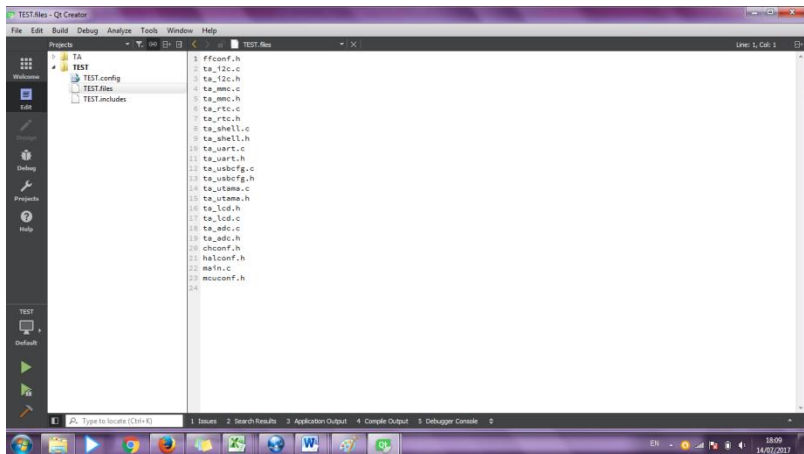
**Gambar 3.19** *import existing project management*

- g. Setelah itu akan muncul *project* yang berisi *file* utama meliputi *.config*, *.files*, *.includes*.



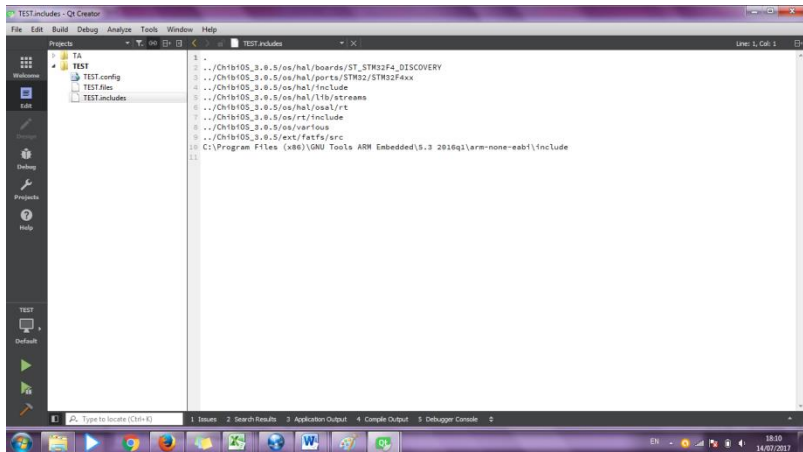
**Gambar 3.20** Tampilan awal program

- h. Kemudian pada project.files diisi fungsi-fungsi yang digunakan sesuai gambar dibawah ini:



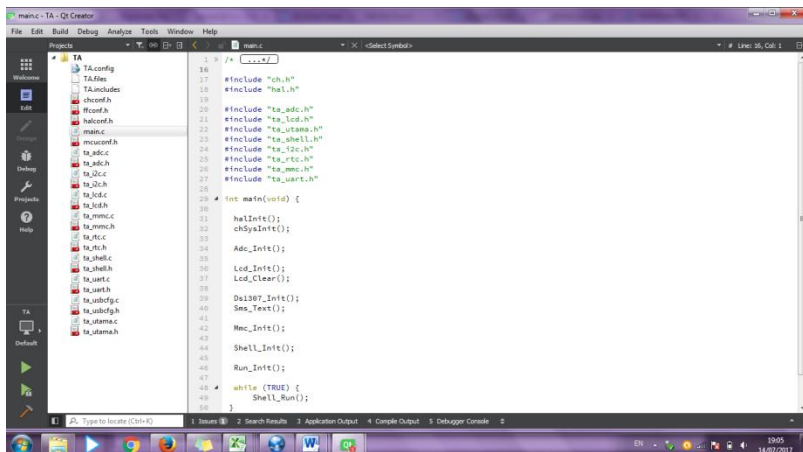
**Gambar 3.21** Program project .files

- i. Pada `project.includes` diisi dengan *library software* yang digunakan yakni ChibiOS seperti gambar dibawah ini:



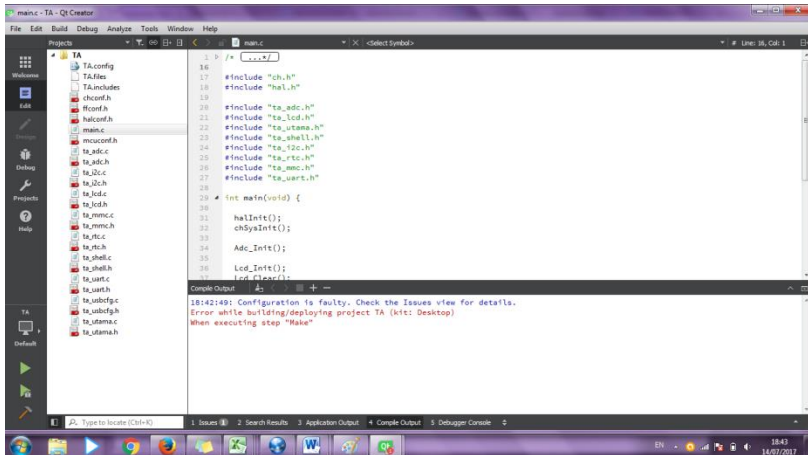
**Gambar 3.22** Program *project.includes*

- j. Setelah konfigurasi telah dibuat, kemudian ditambahkan *source* dan *header* yang digunakan seperti ADC, I2C, MMC, RTC, SHELL, UART, UTAMA, LCD, MAIN, dll



**Gambar 3.23** Class pada *project*

- k. Setelah itu akan muncul program yang telah diatur dan pemrograman bisa dilakukan.
- l. Setelah selesai menyusun program, program dapat di-*compile*.

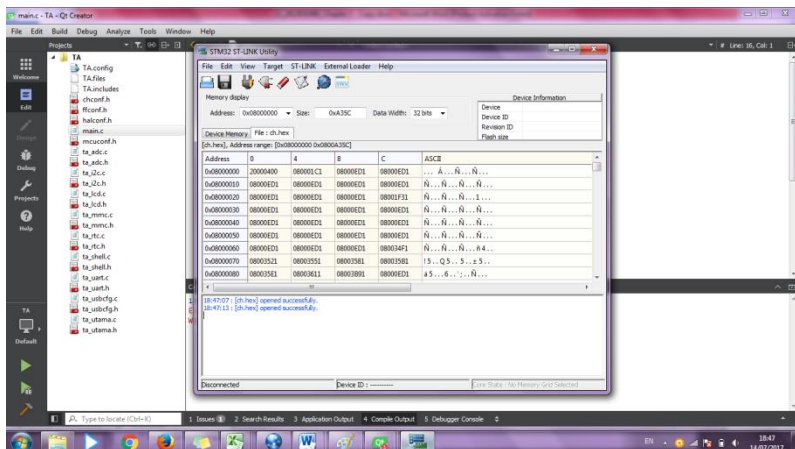


**Gambar 3.24** Build Project Qt Creator

- m. Software Qt Creator dan PC tidak *compatible* sehingga diperlukan aplikasi tambahan berupa *notepad++* untuk meng-*compile*. Pada *notepad++* masukkan *project* yang dibuat (Makefile). Kemudian klik *tools* dan *make all*, seperti gambar dibawah ini:



n. *Download project* yang telah di-compile menggunakan downloader ST-LINK V2 sesuai gambar dibawah ini:



**Gambar 3.26** *Download project di ST-LINK V2*

*(Halaman ini sengaja dikosongkan)*

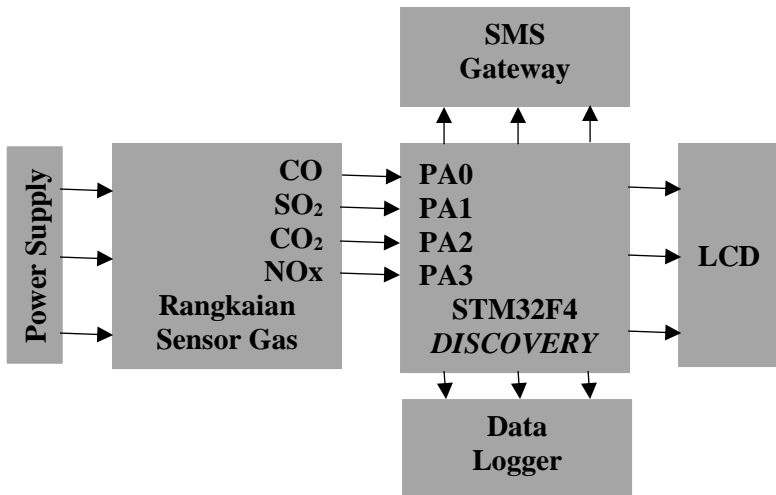
## BAB IV PENGUJIAN DAN ANALISIS DATA

### 4.1 Pengujian Alat

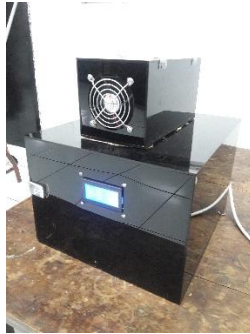
Pada tugas akhir pembuatan “Rancang Bangun Sistem Monitoring Konsentrasi Gas Karbon Dioksida (CO<sub>2</sub>) Sebagai Emisi Gas Buang Menggunakan Sensor MG 811 Berbasis Mikrokontroler *STM32F4 Discovery*” dilakukan pengujian alat serta pengambilan data dilakukan sebagai berikut :

#### 4.1.1 Hasil Rancang Bangun

Berikut pada Gambar 4.1 merupakan alur sistem dari *mini plant* PLTS yang telah dibuat. Rancang bangun system monitoring konsentrasi gas karbon dioksida (CO<sub>2</sub>) ini dirangkai untuk memonitoring konsentrasi gas karbon dioksida (CO<sub>2</sub>)



**Gambar 4.1** Alur Sistem monitoring konsentrasi gas sebagai emisi gas buang



**Gambar 4.2** *Plant monitoring gas*

Pada Gambar 4.2 merupakan sistem monitoring emisi gas buang berupa konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sensor gas MG-811 yang dikontrol oleh mikrokontroler STM32F4 Discovery sebagai pengkondisian sinyal dan tempat pemrosesan sinyal, dan sensor MG 811 sebagai sensor untuk sensing gas. Ketika sensor MG 811 telah mendapat *input* gas maka oleh sensor akan diubah menjadi tegangan dan kemudian *output* ini dibaca oleh ADC (*Analog to Digital Converter*) internal dari mikrokontroler STM32F4 yang kemudian data dikalkulasikan dengan rumusan tertentu sehingga pada tahap berikutnya sistem dapat menentukan apakah tekanan yang dihasilkan sudah sesuai dengan nilai yang seharusnya atau belum. Konsentrasi gas yang terbaca oleh sensor akan ditampilkan ke display LCD (*Liquid Crystal Display*) 16x4 yang ditampilkan dalam satuan ppm.

#### **4.2 Pengujian penyimpanan memori ke SD Card**

Pada pengujian modul SD card ini dilakukan dengan cara pengambilan data yang didapat dari empat sensor (Sistem Monitoring Gas) sensor yang digunakan yaitu sensor MQ 136, sensor MQ 135, sensor MQ 7 dan sensor MG 811. Data yang didapatkan dari sensor akan disimpan ke dalam memori SD card. Data yang disimpan pada SD Card diperoleh setiap 5 detik.

Berikut adalah gambar dari hasil pengujian pembacaan dan penulisan ke dalam memori SD card :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	10/07/2017	15.49	199	438	635	760														
2	10/07/2017	15.49	199	460	637	769														
3	10/07/2017	15.49	199	460	637	752														
4	10/07/2017	15.49	200	461	638	757														
5	10/07/2017	15.49	199	459	635	750														

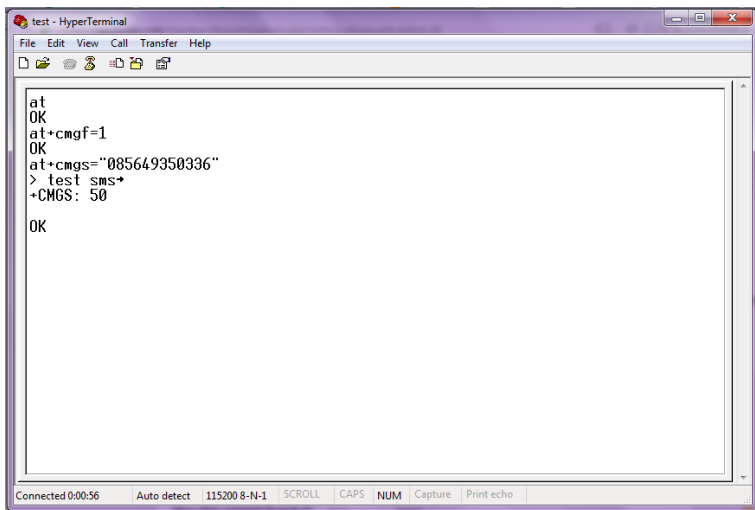
**Gambar 4.3** Pengujian SD Card

Pada gambar 4.3 menjelaskan tentang hasil pembacaan dari empat sensor yang digunakan dan dijelaskan juga kapan waktu pembacaan dan penulisan pada SD card dilakukan. Pada hasil pengujian gambar 4.11 sudah didapatkan hasil yang sesuai dengan yang diinginkan data dapat dibaca atau ditulis kedalam memori SD card dengan hasil yang ditulis berasal dari pembacaan sensor MQ 136, sensor MQ 135, sensor MQ 7 dan sensor MG 811.

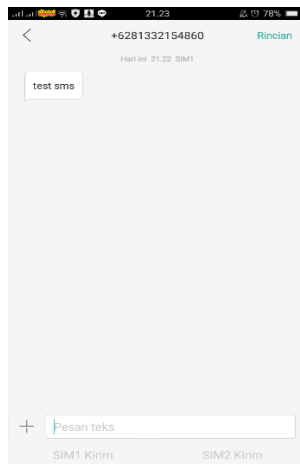
### 4.3 Pengujian Modem untuk SMS Gateway

Pengujian sistem sms gateway ini dilakukan untuk mengetahui apakah program ini mampu mengirim pesan dan menerima pesan. Sistem pengiriman sms diprogram hanya untuk satu nomer telepon. Seperti yang telah dijelaskan sebelumnya bahwa warning system dan monitoring gas CO<sub>2</sub> ini akan mengirimkan sms sebagai peringatan sekaligus pemberitahuan konsentrasi gas CO<sub>2</sub> dan kondisinya apabila melebihi ambang batas yang telah ditetapkan yaitu 600 ppm.

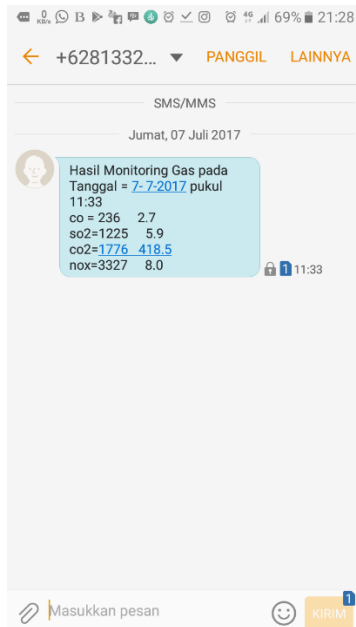
Berikut merupakan hasil pengujian yang telah dilakukan:



**Gambar 4.4** Screenshoot AT Command pada Hyperterminal



**Gambar 4.5** Screenshoot SMS yang diterima pada HP

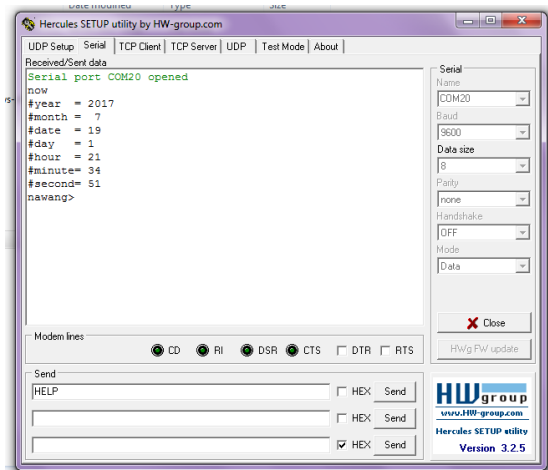


**Gambar 4.6** Screenshot SMS Data Hasil Monitoring

Jadi jika gas CO<sub>2</sub> lebih dari 600 ppm maka akan mengirimkan SMS, jika kondisi SMS sudah terpenuhi maka SMS akan berhenti.

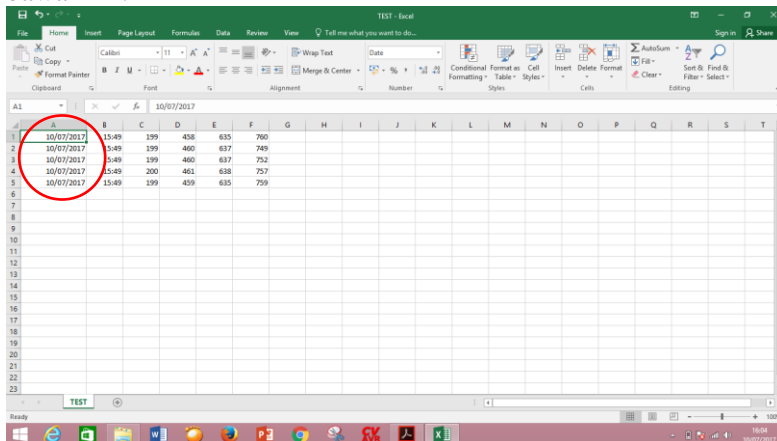
#### 4.4 Pengujian RTC (*Real Time Clock*)

Pengujian rangkaian Real Time Clock dilakukan dengan menerapkan program pada mikrokontroler STM32F4 *Discovery*, dimana RTC DS3231 sebagai timer, dan dibutuhkan tegangan 5V untuk mengaktifkan rangkaian RTC DS3231. Untuk mengatur jam dan tanggal pada RTC dilakukan pengujian melalui Hyperteminal. Bisa dilihat pada gambar 4.10 dibawah ini,



**Gambar 4.7** Pengujian data waktu RTC pada Hyperterminal

Pada saat pengujian terlihat bahwa RTC DS3231 bisa bekerja dengan baik yaitu bisa menampilkan jam dan tanggal sesuai dengan program yang sudah dimasukan sebelumnya. Data waktu yang sudah sesuai dicantumkan ke format excel yang akan disimpan ke dalam SD Card. Bisa dilihat pada Gambar 4.11 di bawah ini.



**Gambar 4.8** Pengujian data waktu



#### 4.5 Pengujian Sensor Gas MG - 811

Berikut merupakan hasil pengujian sensor gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sensor MG 811. Pengambilan data dilakukan berdasarkan variansi waktu berbasis STM32F4 *Discovery*.



**Gambar 4.9** Peletakan Sensor  $\text{CO}_2$

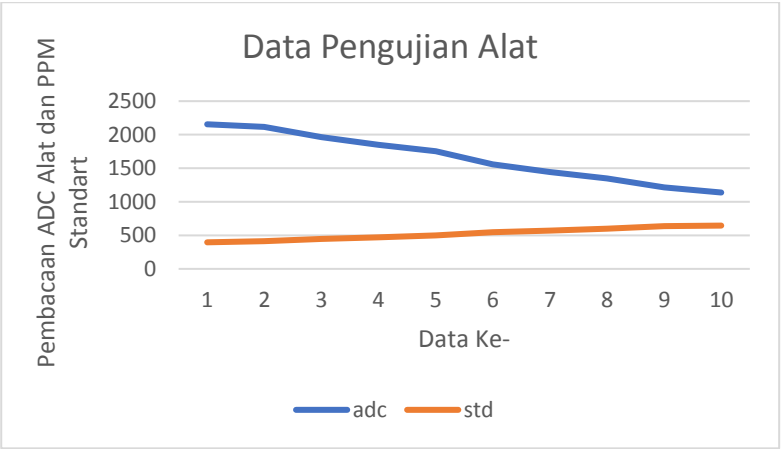
Pada alat *monitoring* emisi gas buang ini terdapat 4 sensor yaitu sensor  $\text{CO}$ ,  $\text{SO}_2$ ,  $\text{CO}_2$ ,  $\text{NO}_x$ . Sensor yang digunakan pada gas karbon dioksida ( $\text{CO}_2$ ) ini yaitu MG – 811 dimana sensor ini memiliki *range* pengukuran sebesar 350ppm – 10000ppm.

Pengujian dilakukan dengan mengaktifkan tegangan referensi 5 V yang masuk ke *sensor* kemudian diberi input berupa sampel gas uji dan keuaran dari sensor berupa sinyal analog akan diproses menjadi sinyal digital dalam mikrokontroler STM32F4 *Discovery*.

Berikut ini data yang diperoleh dari pengujian alat dan grafiknya pada gambar 4.2

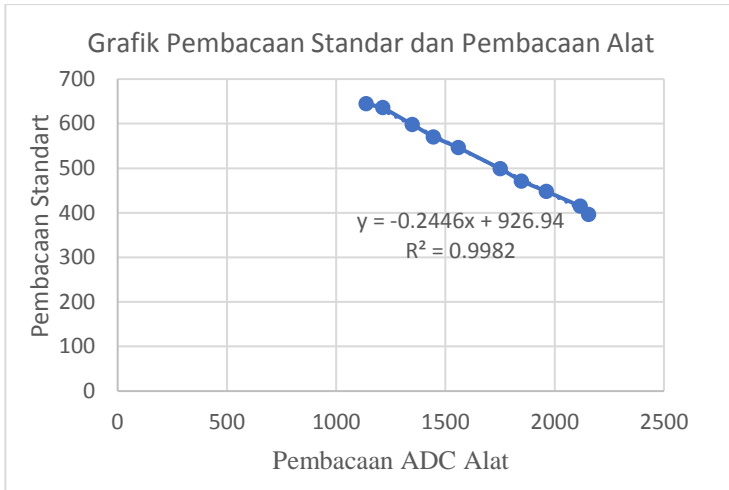
**Tabel 4.1** Data Pengujian Alat

No.	Pembacaan Standar (PPM)	Pembacaan ADC Alat
1	645	2155
2	636	2117
3	598	1963
4	570	1848
5	546	1752
6	499	1560
7	471	1445
8	448	1349
9	415	1214
10	396	1138



**Gambar 4.10** Grafik Pembacaan Standar dan Pembacaan Alat

Jika tipe grafik diubah menjadi tipe scatter, maka akan didapatkan grafik seperti dibawah ini:



**Gambar 4.11** Grafik Pembacaan Standar dan Pembacaan ADC Alat

Dari data diatas diperoleh grafik perbandingan perbedaan antara alat monitoring dengan alat standar. Alat standar yang digunakan adalah Impinger. Karena adanya perbedaan hasil maka dilakukan perhitungan persamaan regresi dengan rumus sebagai berikut:

$$Y = a + bx$$

Dimana,

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

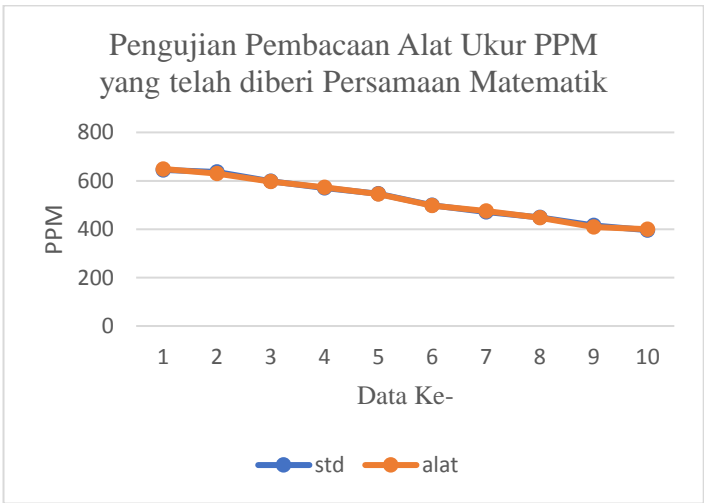
$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Dari perhitungan diatas diperoleh hasil  $y = -0.2446x + 926.94$  dimana y sebagai nilai dari pembacaan alat standar dan x sebagai nilai pembacaan alat. Persamaan matematik tersebut menghasilkan data pembacaan alat ppm karbon dioksida ( $\text{CO}_2$ ) yang sudah mendekati dengan pembacaan alat standar, impinger karbon dioksida ( $\text{CO}_2$ ). Maka data yang digunakan seperti berikut:

**Tabel 4.2** Tabel Konversi Data ADC ke PPM

No	Kadar Gas (ppm)	ADC	Nilai ppm alat
1	645	2155	649
2	636	2117	630
3	598	1963	597
4	570	1848	573
5	546	1752	545
6	499	1560	498
7	471	1445	475
8	448	1349	447
9	415	1214	409
10	396	1138	400

Berikut adalah grafik pengujian pembacaan alat ukur ppm yang telah diberi persamaan matematik



**Gambar 4.12** Grafik Pembacaan Standar dan Pembacaan Alat dengan persamaan matematik

#### 4.6 Data Spesifikasi Alat

Karakteristik statik adalah karakteristik yang harus diperhatikan apabila alat tersebut digunakan untuk mengukur suatu kondisi yang tidak berubah karena waktu atau hanya berubah secara lambat laun. Untuk itu perlu dilakukan perhitungan untuk mengetahui nilai karakteristik dari *sensor* MG 811 diantaranya sebagai berikut:

**Tabel 4.3** Pengambilan Data pada *Sensor* MG 811

No	Standart	Alat	(STD-alat)/std	Non linearitas
1	396	400	-0.010	0.000
2	415	409	0.014	-10.000
3	448	447	0.002	-5.000
4	471	475	-0.008	0.000
5	499	498	0.002	-5.000
6	546	545	0.002	-5.000
7	570	573	-0.005	-1.000
8	598	597	0.002	-5.000
9	636	630	0.009	-10.000
10	645	649	-0.006	0.000
Jumlah	5224	5223.00	0,00	
Rata-rata	522.4	522.30	0,00	

Sehingga dihasilkan nilai:

*Range* : 400 – 649 ppm  
*Span* : 249  
 Resolusi : 0,01  
 Non-linieritas : 0 %  
 Akurasi : 99%  
 Kesalahan (*Error*) : 1%

Berikut ini hasil perhitungan nilai karakteristik statik *pressure* berdasarkan data pada tabel 4.3:

- a. Non – Linieritas ( $N(I)$ ) =  $O(I) - (KI + a)$

\*(Berdasarkan data naik)

Non – Linieritas maksimum per unit

$$= \frac{\hat{N}}{O_{max} - O_{min}} \times 100\%$$

Dimana:

$$K \text{ (sensitifitas)} = \frac{\Delta O}{\Delta I} = \frac{649 - 400}{645 - 396} = 1$$

$$a \text{ (zero bias)} = O_{min} - KI_{min}$$

$$a = 400 - (1 \times 396)$$

$$a = 4$$

Sehingga:

Non – Linieritas maks. Per unit

$$= \frac{0}{649 - 400} \times 100\%$$

$$\text{Non – linieritas} = 0\%$$

- b. Akurasi:

$$A = 1 - \sum \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%$$

Dengan:

$Y_n$  = Pembacaan standar (I) dan

$X_n$  = Pembacaan alat (O)

$$A = 1 - (0.010) \times 100\%$$

$$A = 0.99 \times 100\%$$

$$A = 99\%$$

- c. Error:

$$e = 1 - A$$

$$e = 1 - 0.99$$

$$e = 0.010 \times 100\% = 1\%$$

Berikut ini merupakan hasil pengukuran kalibrasi untuk mencari nilai ketidakpastian alat ukur, dimana kalibrasi dilakukan di ruangan terbuka:

**Tabel 4.4** Data Kalibrasi pada *Sensor MG 811 (1)*

No	Pembacaan Standart	Pembacaan Alat	Koreksi (Y)	$\text{xi}^2$
1	396	400	-4.00	156816
2	415	409	6.00	172225
3	448	447	1.00	200704
4	471	475	-4.00	221841
5	499	498	1.00	249001
6	546	545	1.00	298116
7	570	573	-3.00	324900
8	598	597	1.00	357604
9	636	630	6.00	404496
10	645	649	-4.00	416025
Jumlah	5224	5223.00	1.00	2801728
Rata-rata	522.4	522.30	0.10	

**Tabel 4.5** Data Kalibrasi pada *Sensor* MG 811 (2)

No	xi*yi	Yreg	R	SSR
1	-1584.00	-0.102697826	-3.90	15.19
2	2490.00	-0.072229007	6.07	36.87
3	448.00	-0.01930948	1.02	1.04
4	-1884.00	0.017573827	-4.02	16.14
5	499.00	0.062475244	0.94	0.88
6	546.00	0.13784548	0.86	0.74
7	-1710.00	0.176332409	-3.18	10.09
8	598.00	0.221233826	0.78	0.61
9	3816.00	0.282171464	5.72	32.69
10	-2580.00	0.296604062	-4.30	18.46
Jumlah	639.00			132.71

- a. Nilai Ketidakpastian *Type A*:

$$\sigma \text{ (Standar Deviasi)} = \frac{\sqrt{\sum (y_i - \bar{y})^2}}{n-1}$$

Sehingga nilai ketidakpastian hasil pengukuran:

$$U_{a1} = \frac{\sigma}{\sqrt{n}}$$

$$U_{a1} = \frac{1.0485}{\sqrt{0}} = 0$$

Sedangkan nilai ketidakpastian regresi  $U_{a2} = \sqrt{\frac{SSR}{n-2}}$

Dimana:

SSR (*Sum Square Residual*) =  $\sum SR$  (*Square Residual*)

SR =  $R^2$  (*Residu*)

$Y_i$  = Nilai koreksi ke-i

$$Y_{reg} = a + (b \cdot x_i)$$

$$a = \bar{y}_i + (b \cdot \bar{x}_i)$$



$$b = \frac{n \cdot \sum x_i y_i - \sum x_i \sum y_i}{n \cdot \sum x_i^2 - (\sum x_i)^2}$$

Dimana:

$x_i$  = Pemb. standar,

$y_i$  = Nilai koreksi,  $n$  = Jumlah data

$$b = \frac{(6390) - (5224)}{(28017280) - (27290176)} = 0,001603622$$

Sehingga nilai:

$$a = 0.10 + (0,0016 \times 522,4)$$

$$a = 0.93584$$

Jadi, persamaan regresi menjadi:

$$Y_{reg} = (0.10) + (x_i \times (0,0016))$$

Yang menghasilkan nilai  $SSR = 132,71$

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{132,71}{10-2}} = 4,07293$$

b. Nilai Ketidakpastian Type B :

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi ( $U_{b1}$ ) dan Ketidakpastian alat standar pressure gauge ( $U_{b2}$ ). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{a}{\sqrt{3}} = \frac{0.93584}{\sqrt{3}} = 0,5403$$

$$U_{b2} = \frac{\frac{1}{2} \times \text{Resolusi}}{\sqrt{3}} = \frac{\frac{1}{2} \times 0,01}{\sqrt{3}} = 0.003,$$

- c. Nilai ketidakpastian kombinasi  $U_c$ :

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2}$$

$$U_c = \sqrt{0^2 + 4,0729^2 + 0,5403^2 + 0,003^2}$$

$$U_c = 4,1085$$

Dengan kondisi V atau derajat kebebasan dari kedua tipe ketidakpastian, sebagai berikut :

$V = n-1$ , sehingga :

$V1 = 9$ ;  $V2 = 9$ ;  $V3 = 0,5$ ;  $V4 = 0,5$  (berdasarkan table *T-Student*)

Dengan nilai  $V_{eff}$  (Nilai derajat kebebasan efektif) sebagai berikut :

$$V_{eff} = \frac{(U_c)^4}{\sum (U_i)^4 / V_i}$$

$V_{eff}$

$$= \frac{(4,1085)^4}{(0)^4/9 + (4,07293)^4/9 + (0,5403)^4/0,5 + (0,003)^4/0,5}$$

$$V_{eff} = 9,2668940364,$$

Setelah mendapatkan nilai  $V_{eff}$  dapat digunakan untuk mencari factor cakupan (k), digunakan rumus interpolasi terdekat yaitu sebagai berikut:

$$\frac{X - X1}{X2 - X1} = \frac{Y - Y1}{Y2 - Y1}$$

$$\begin{aligned}
 \frac{9,2 - 9}{10 - 9} &= \frac{Y - 1,83}{1,81 - 1,83} \\
 \frac{0,2}{1} &= \frac{Y - 1,83}{-0,02} \\
 -0,004 &= Y - 1,83 \\
 1,83 - 0,004 &= Y \\
 1,826 &= Y
 \end{aligned}$$

Sehingga didapatkan nilai ketidakpastian diperluang sebesar adalah sebagai berikut:

$$U_{exp} = k \times U_c$$

$$U_{exp} = 1,826 \times 4,1085 = 7,502121$$

Setelah dilakukan semua perhitungan, dapat dilihat hasil pengukuran sebagai berikut :

X	= $\bar{X} \pm U_{exp}$
X	= $(522,3 \pm 7,502121)\text{ppm}$
CL(Confidence Level)	= 95%
k	= $\pm 1,826$

#### 4.7 Pembahasan

Sistem monitoring emisi gas buang berupa konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) menggunakan sensor gas MG-811 yang dikontrol oleh mikrokontroler STM32F4 Discovery sebagai pengkondisian sinyal dan tempat pemrosesan sinyal, dan sensor MG 811 sebagai sensor untuk sensing gas. Ketika sensor MG 811 telah mendapat *input* gas maka oleh sensor akan diubah menjadi tegangan dan kemudian *output* ini dibaca oleh ADC (*Analog to Digital Converter*) internal dari mikrokontroler STM32F4 yang kemudian data dikalkulasikan dengan rumusan tertentu sehingga pada tahap berikutnya sistem dapat menentukan apakah tekanan yang dihasilkan sudah sesuai dengan nilai yang seharusnya atau belum. Konsentrasi gas yang terbaca oleh sensor akan ditampilkan

ke display LCD (*Liquid Crystal Display*) 16x4 yang ditampilkan dalam satuan ppm.

Prinsip kerja dari sensor karbon dioksida yaitu bila mana sensor terkena gas, terutama gas karbon dioksida akan merespon dengan baik. Sensor gas karbon dioksida akan mengeluarkan tegangan sebesar konsentrasi gas yang dideteksinya. Berdasarkan pengkondisian sinyal yaitu modul sensor gas karbon dioksida (MG 811). Dan selanjutnya akan dibaca oleh STM32F4 Discovery. Setelah itu mikrokontroler akan mengambil tindakan untuk menampilkan besarnya konsentrasi gas karbon dioksida pada LCD dalam satuan PPM, dan juga menyimpan data pada sd card sekaligus mengirimkan sms jika konsentrasi gas tersebut sudah melewati batas ambang.

Dapat dilihat Sistem *Monitoring* gas untuk pengukuran gas CO<sub>2</sub> memiliki nilai ketidakpastian  $U_{a1} = 0$ ;  $U_{a2} = 4,07293$ ;  $U_{b1} = 0,5403$  dan  $U_{b2}$  sebesar 0,003, alat ukur memiliki nilai ketidakpastian diperluas ( $U_{exp}$ ) sebesar  $\pm 7,502121$  dengan tingkat kepercayaan 95%

Setelah melakukan perhitungan dapat diketahui bahwa Nilai ketidakpastian pengukuran alat ukur 1.0485 , Nilai hasil pengukuran  $x = (522,3 \pm 7,502121)$ ppm. *Confidence level* (CL) 95%, dan factor cakupan (k) sebesar  $\pm 1,826$ .

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan pada hasil penelitian tugas akhir yang sudah dilakukan dapat disimpulkan sebagai berikut:

1. Telah berhasil dibuat rancang bangun sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang dengan range pengukuran sebesar 400-649 ppm, resolusi sebesar 0,01, Non-linieritas sebesar 0 %, Akurasi sebesar 99%, dan Kesalahan (*Error*) sebesar 1%.
2. Hasil kalibrasi dari sistem monitoring konsentrasi gas karbon dioksida ( $\text{CO}_2$ ) sebagai emisi gas buang didapatkan nilai hasil pengukuran  $x = (522,3 \pm 7,502121)\text{ppm}$ . *Confidence level* (CL) 95%, dan factor cakupan (k) sebesar  $\pm 1,826$ .
3. Penggunaan data hasil monitoring gas karbon dioksida ( $\text{CO}_2$ ), dilakukan dengan cara penyimpanan data logger di SD card dengan cara penyimpanan dengan rule yang ditentukan dan mengirimkan sms gateway apabila salah satu parameter gas melebihi kondisi normal.

#### **5.2 Saran**

Saran yang diberikan untuk dilakukan penelitian selanjutnya yaitu:

1. Dibutuhkan sampel gas uji yang dapat mengatur range ppm dengan baik, sehingga dapat melakukan hasil kalibrasi yang akurat.
2. Lakukan perbandingan dengan *gas detector* yang sudah ada agar dapat menjadi perbandingan atau acuan nilai dari hasil pengukuran gas.
3. Gunakan sumber gas yang murni agar didapatkan hasil pendeteksian yang lebih baik.

## DAFTAR PUSTAKA

- [1] <http://pollutiononmyearth.weebly.com/pencemaran-udara.html> (Diakses 5 Juni 2017, 20.20 WIB).
- [2] <http://www.organisasi.org/1970/01/pencemaran-udara-pada-lingkungan-hidup-sekitar-kita-gas-beracun-co-co2-no-no2-so-dan-so2-yang-merusak-kesehatan-manusia.html> (Diakses 5 Juni 2017, 20.20 WIB)
- [3] <http://www.solopos.com/2009/10/16/surabaya-peringkat-ketiga-kota-berpolusi-di-asia-6351> (Diakses 07 Juni 2017, 13.00 WIB)
- [4] C. Davidson, "Marine Notice: Carbon Dioxide: Health Hazard". Australian Maritime Safety Authority, 7 February 2003..
- [5] Occupational Safety and Health Administration, Chemical Sampling Information: "Carbon Dioxide", tersedia di : [http://www.osha.gov/dts/chemicalsampling/data/CH\\_225400.html](http://www.osha.gov/dts/chemicalsampling/data/CH_225400.html). (Diakses 07 Juni 2017, 13.00 WIB)
- [6] Rice A.S., 2003. Health effects of acute and prolonged CO<sub>2</sub> exposure in normal and sensitive populations presented at second annual conference on carbon sequestration. Virginia, USA. 5-8 May.
- [7] Wildan Aryo Wicaksono, Indra Astriawan Suseno, Perancangan Sistem Monitoring Tingkat Pencemaran Udara Dengan Komunikasi SMS Menggunakan Modem GSM, **Tugas Akhir**, D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, 2014.
- [8] STMicroelectronics. 2012. *STM32F4DISCOVERY, STM32F4 high performance discovery board*
- [9] Hanwei., 2015. MG 811 CO<sub>2</sub> Sensor. Access Date: August 20th, 2015.
- [10] Michael D. Robinson, Gary Sharp, dan Jianmin Chen, **Polarization Engineering For LCD Projection**, John Wiley & Sons Ltd., 2005.
- [11] <http://www.ngarep.net/tutorial-arduino-mengakses-modul-micro-sd/> (Diakses 8 Juni 2017, 23.00 WIB)

- [12] <http://www.indo-ware.com/produk-3048-ds3231-at24c32-clock-module.html> (Diakses 8 Juni 2017, 23.00 WIB)
- [13] <http://repository.usu.ac.id/bitstream/handle/123456789/42752/Chapter%20II.pdf;jsessionid=8B937B36E42975C825EE710A703ACA83?sequence=3> (Diakses 8 Juni 2017, 23.00 WIB)
- [14] Kho, Dickson. 2017. Pengertian MCB (Miniature Circuit Breaker) dan Prinsip Kerjanya. <http://teknikelektronika.com/pengertian-mcb-miniature-circuit-breaker-prinsip-kerja-mcb/> (Diakses 10 Juni 2017, 13.00 WIB).
- [15] Atmaja, Bagus. 2015. Rancang Bangun STM32 dengan ChibiOs. <http://www.bagustris.tk/2015/04/rancang-bangun-modul-stm32-dengan.html> (Diakses 10 Juni 2017, 13.00 WIB).
- [16] <http://www.chibios.org/dokuwiki/doku.php> (Diakses 10 Juni 2017, 13.00 WIB).
- [17] Laboratorium Pengukuran Fisis. 2013. Modul Teknik Pengukuran dan Kalibrasi. Teknik Fisika, FFI-ITS, Surabaya.
- [18] <https://ask.fm/AdjengIS/answers/116659751914> (Diakses 17 Juni 2017, 19.00 WIB).
- [19] [http://www.proweb.co.id/articles/mobile\\_development/qt\\_creator.html](http://www.proweb.co.id/articles/mobile_development/qt_creator.html) (Diakses 10 Juni 2017, 13.00 WIB).
- [20] <http://blog.dimensidata.com/pengertian-dan-perbedaan-kualitas-class-pada-microsd/>. (Diakses 17 Juni 2017, 19.00 WIB).
- [21] <http://bisakomputer.com/notepad-text-editor-keren-serba-guna/> (Diakses 17 Juni 2017, 19.00 WIB).
- [22] ..., "Seputar Modem Wavecom Fastrack". <http://kiswara.com/seputar-modemwavecom-fastrack-67-19.info> (Diakses 17 Juni 2017, 19.00 WIB, 13.00 WIB).
- [23] Bentley, John P. *Principles of Measurement Systems – 4th edition*. London : Pearson Education Limited.





## **LAMPIRAN A**

### **(*LISTING PROGRAM PADA CHIBIOS/RT*)**

/\*\*\*\*\*\*

ChibiOS/RT - Copyright (C) 2006-2013 Giovanni Di Sirio

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

\*\*\*\*\*/

**//main.c**

#include "ch.h"

#include "hal.h"

#include "ta\_adc.h"

#include "ta\_lcd.h"

#include "ta\_utama.h"

#include "ta\_shell.h"

#include "ta\_i2c.h"

#include "ta\_rtc.h"

#include "ta\_mmc.h"

#include "ta\_uart.h"

```

int main(void) {
    halInit();
    chSysInit();
    Adc_Init();
    Lcd_Init();
    Lcd_Clear();
    Ds1307_Init();
    Sms_Text();
    Mmc_Init();
    Shell_Init();
    Run_Init();
    while (TRUE) {
        Shell_Run();
    }
}

```

**//adc.c**

```
#include "ta_adc.h"
```

```

static adcsample_t samples[ADC_GRP1_NUM_CHANNELS *
    ADC_GRP1_BUF_DEPTH];
adcsample_t adc0,adc1,adc2,adc3;
uint32_t sum_adc0,sum_adc1,sum_adc2,sum_adc3;
void adccb(ADCDriver *adcp, adcsample_t *buffer, size_t n){
    (void) buffer; (void) n;
    int i;
    if (adcp->state == ADC_COMPLETE) {
        sum_adc0=0;
        sum_adc1=0;
        sum_adc2=0;
        sum_adc3=0;
        for(i=0;i<ADC_GRP1_BUF_DEPTH;i++){
            sum_adc0=sum_adc0+samples[0+(i*ADC_GRP1_NUM_
                CHANNELS)];
            sum_adc1=sum_adc1+samples[1+(i*ADC_GRP1_NUM_
                CHANNELS)];

```

```

        sum_adc2=sum_adc2+samples[2+(i*ADC_GRP1_NUM_CHANNELS)];
        sum_adc3=sum_adc3+samples[3+(i*ADC_GRP1_NUM_CHANNELS)];
    }
    adc0=sum_adc0/ADC_GRP1_BUF_DEPTH;
    adc1=sum_adc1/ADC_GRP1_BUF_DEPTH;
    adc2=sum_adc2/ADC_GRP1_BUF_DEPTH;
    adc3=sum_adc3/ADC_GRP1_BUF_DEPTH;
}
}
static const ADCConversionGroup adcgprcfg = {
    FALSE,
    ADC_GRP1_NUM_CHANNELS,
    adccb,
    NULL,
    /* HW dependent part.*/
    0,
    ADC_CR2_SWSTART,
    0,
    ADC_SMPR2_SMP_AN0(ADC_SAMPLE_112) |
        ADC_SMPR2_SMP_AN1(ADC_SAMPLE_112) |
        ADC_SMPR2_SMP_AN2(ADC_SAMPLE_112) |
        ADC_SMPR2_SMP_AN3(ADC_SAMPLE_112),
    ADC_SQR1_NUM_CH(ADC_GRP1_NUM_CHANNELS),
    0,
    ADC_SQR3_SQ1_N(ADC_CHANNEL_IN0) |
        ADC_SQR3_SQ2_N(ADC_CHANNEL_IN1) |
        ADC_SQR3_SQ3_N(ADC_CHANNEL_IN2) |
        ADC_SQR3_SQ4_N(ADC_CHANNEL_IN3)
};
static THD_WORKING_AREA(wa_adcThread, 128);
static THD_FUNCTION(adcThread, arg) {
    (void)arg;
    chRegSetThreadName("ADC Run");
    while (TRUE) {

```

```

    chThdSleepMilliseconds(100);
    palSetPad(GPIOD, 12); /* Yellow. */
    adcStartConversion(&ADCD1,      &adcgrpcfg,      samples,
        ADC_GRP1_BUF_DEPTH);
    chThdSleepMilliseconds(100);
    palClearPad(GPIOD, 12); /* Yellow. */
}
}
void Adc_Init(){
    palSetPadMode(GPIOA,0,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,1,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,2,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,3,PAL_MODE_INPUT_ANALOG);
    adcStart(&ADCD1, NULL);
    adcSTM32EnableTSVREFE();
    palSetPadMode(GPIOD,12,PAL_MODE_OUTPUT_PUS
        HPULL);
    chThdCreateStatic(wa_adcThread,      sizeof(wa_adcThread),
        NORMALPRIO, adcThread, NULL);
}

```

**//adc.h**

```

#ifndef TA_ADC_H
#define TA_ADC_H

```

```

#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <stdarg.h>
#include <stdlib.h>
#include <math.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"

```

```

#ifndef CHPRINTF_USE_FLOAT
#define CHPRINTF_USE_FLOAT FALSE
#endif
#define MAX_FILLER 16
#define FLOAT_PRECISION 100
#define ADC_GRP1_NUM_CHANNELS 4
#define ADC_GRP1_BUF_DEPTH 100
void Adc_Init(void);
#endif

//i2c.c
#include "ta_i2c.h"

static const I2CConfig i2cconfig= {
    OPMODE_I2C,
    400000,
    FAST_DUTY_CYCLE_2,
};
uint8_t readByteI2C(uint8_t addr){
    uint8_t data;
    i2cAcquireBus(&I2CD1);
    (void)
        i2cMasterReceiveTimeout(&I2CD1,addr,&data,1,TIME_I
            NFINITE);
    i2cReleaseBus(&I2CD1);
    return data;
}
void writeByteI2C(uint8_t addr, uint8_t reg, uint8_t val){
    uint8_t cmd[] = {reg, val};
    i2cAcquireBus(&I2CD1);
    (void) i2cMasterTransmitTimeout(&I2CD1, addr, cmd, 2,
        NULL, 0, TIME_INFINITE);
    i2cReleaseBus(&I2CD1);
}
void I2c_Init(void){

```

```

    palSetPadMode(GPIOB,8,PAL_MODE_ALTERNATE(4) |
        PAL_STM32_OTYPE_OPENDRAIN);
    palSetPadMode(GPIOB,9,PAL_MODE_ALTERNATE(4) |
        PAL_STM32_OTYPE_OPENDRAIN);
    i2cStart(&I2CD1, &i2cconfig);
}

```

### //i2c.h

```

#ifndef TA_I2C_H
#define TA_I2C_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
uint8_t readByteI2C(uint8_t addr);
void writeByteI2C(uint8_t addr, uint8_t reg, uint8_t val);
void I2c_Init(void);
#endif

```

### //lcd.c

```

#include "ta_lcd.h"

LcdStream myLCD;
static msg_t put(void *ip, uint8_t chr) {
    (void)ip;
    Lcd_Write_Data(chr);
    return MSG_OK;
}
static const struct LcdStreamVMT vmt = {NULL, NULL, put,
    NULL};
void lsObjectInit(LcdStream *msp) {

```

```

msp->vmt = &vmt;
}
void Lcd_Pin_Dir(void){
    palSetPadMode(LCD_PORT_CRTL,LCD_PIN_RS,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_CRTL,LCD_PIN_EN,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D4,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D5,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D6,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D7,LCD_
PORT_MODE);
}
void Lcd_Write_Data(uint8_t chr){
    palWritePort(LCD_PORT_DATA,(chr & 0xf0));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_RS);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(10);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_RS);
    chThdSleepMilliseconds(10);

    palWritePort(LCD_PORT_DATA,((chr & 0x0f)<<4));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_RS);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(10);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_RS);
    chThdSleepMilliseconds(10);
}
void Lcd_Write_Command(uint8_t cmd){
    palWritePort(LCD_PORT_DATA,(cmd & 0xf0));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);

```

```

chThdSleepMilliseconds(10);
palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);

palWritePort(LCD_PORT_DATA,((cmd & 0x0f)<<4));
palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);
palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);
}
void Lcd_Cursor(uint8_t column, uint8_t line){
    uint8_t position = 0x00;
    if(column>=TLCD_MAXX) column=0;
    if(line>=TLCD_MAXY) line=0;
    switch(line)
    {
        case 0: position = LCD_LINE0_DDRAMADDR+column;
            break;
        case 1: position = LCD_LINE1_DDRAMADDR+column;
            break;
        case 2: position = LCD_LINE2_DDRAMADDR+column;
            break;
        case 3: position = LCD_LINE3_DDRAMADDR+column;
            break;
    }
    Lcd_Write_Command(1<<LCD_DDRAM | position);
}
void Lcd_Init(void){
    lsObjectInit(&myLCD);
    Lcd_Pin_Dir();
    chThdSleepMilliseconds(500);
    palWritePort(LCD_PORT_CRTL,0x00);
    palWritePort(LCD_PORT_DATA,0x00);

    palSetPad(LCD_PORT_DATA,LCD_PIN_D5);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);

```



```

    chThdSleepMilliseconds(40);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(40);

    Lcd_Write_Command(0x28);
    chThdSleepMilliseconds(10);
    Lcd_Write_Command(0x0c);
    chThdSleepMilliseconds(10);
}
void Lcd_Clear (void){
    Lcd_Write_Command(0x01);
    chThdSleepMilliseconds(10);
}
void Lcd_Example(){
    Lcd_Clear();
    Lcd_Cursor(0,0);
    chprintf((BaseSequentialStream *)&myLCD,"A-LCD");
    Lcd_Cursor(0,1);
    chprintf((BaseSequentialStream *)&myLCD,"Works");
    Lcd_Cursor(0,2);
    chprintf((BaseSequentialStream *)&myLCD,"horee");
    Lcd_Cursor(0,3);
    chprintf((BaseSequentialStream *)&myLCD,"yeee");
}

```

### //lcd.h

```

#ifndef TA_LCD_H
#define TA_LCD_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"

```

```

#include "chstreams.h"
#define _lcd_stream_data _base_sequential_stream_data
#define LCD_PIN_RS 0
#define LCD_PIN_EN 1
#define LCD_PORT_CTRL GPIOE
#define LCD_PIN_D4 4
#define LCD_PIN_D5 5
#define LCD_PIN_D6 6
#define LCD_PIN_D7 7
#define LCD_PORT_DATA GPIOE
#define                                LCD_PORT_MODE
    PAL_MODE_OUTPUT_PUSHPULL
#define TLCD_MAXX      16 // max x-Position (0...15)
#define TLCD_MAXY      4 // max y-Position (0...1)
#define LCD_DDRAM      7
#define LCD_LINE0_DDRAMADDR      0x00
#define LCD_LINE1_DDRAMADDR      0x40
#define LCD_LINE2_DDRAMADDR      0x10
#define LCD_LINE3_DDRAMADDR      0x50

struct LcdStreamVMT {
    _base_sequential_stream_methods
};
typedef struct {
    const struct LcdStreamVMT *vmt;
    _base_sequential_stream_data
} LcdStream;

#ifdef __cplusplus
extern "C" {
#endif
void lsObjectInit(LcdStream *msp);
#ifdef __cplusplus
}
#endif

```

```

void Lcd_Pin_Dir(void);
void Lcd_Write_Command(uint8_t cmd);
void Lcd_Write_Data(uint8_t chr);
void Lcd_Init(void);
void Lcd_Cursor(uint8_t column, uint8_t line);
void Lcd_Clear (void);
void Lcd_Example (void);
#endif // LIB_LCD_H

```

**//mmc.c**

```
#include "ta_mmc.h"
```

```

FATFS MMC_FS;
MMCDriver MMCD1;
bool fs_ready = FALSE;
FRESULT err;
uint32_t clusters;
FATFS *fsp;
uint8_t fbuff[1024];
static SPIConfig hs_spicfg = {NULL, GPIOB, 12, 0};
static SPIConfig ls_spicfg = {NULL, GPIOB, 12, SPI_CR1_BR_2
    | SPI_CR1_BR_1};
static MMCCConfig mmccfg = {&SPID2, &ls_spicfg,
    &hs_spicfg};
FRESULT scan_files(BaseSequentialStream *chp, char *path) {
    FRESULT res;
    FILINFO fno;
    DIR dir;
    int i;
    char *fn;

    #if _USE_LFN
        fno.lfname = 0;
        fno.lfsize = 0;
    #endif
    res = f_opendir(&dir, path);

```

```

if (res == FR_OK) {
    i = strlen(path);
    for (;;) {
        res = f_readdir(&dir, &fno);
        if (res != FR_OK || fno.fname[0] == 0)
            break;
        if (fno.fname[0] == '.')
            continue;
        fn = fno.fname;
        if (fno.fattrib & AM_DIR) {
            path[i++] = '/';
            strcpy(&path[i], fn);
            res = scan_files(chp, path);
            if (res != FR_OK)
                break;
            path[--i] = 0;
        }
        else {
            chprintf(chp, "%s/%s\r\n", path, fn);
        }
    }
}
return res;
}

void Mmc_Mount(void) {
    if (fs_ready) {
        return;
    }
    if (mmcConnect(&MMCD1)) {
        return;
    }
    err = f_mount(&MMC_FS, "/", 1);
    if (err != FR_OK) {
        mmcDisconnect(&MMCD1);
        fs_ready = FALSE;
        return;
    }
}

```

```

    }
    fs_ready = TRUE;
}

void Mmc_Unmount(void) {
    f_mount(NULL,"/",1);
    mmcDisconnect(&MMCD1);
    fs_ready = FALSE;
}

FRESULT f_append (
    FIL* fp,          /* [OUT] file object to create */
    const char* path  /* [IN]  file name to be opened */
)
{
    FRESULT fr;
    /* Opens an existing file. If not exist, creates a new file. */
    fr = f_open(fp, path, FA_WRITE | FA_OPEN_ALWAYS |
        FA_READ);
    if (fr == FR_OK) {
        /* Seek to end of the file to append data */
        fr = f_lseek(fp, f_size(fp));
        if (fr != FR_OK)
            f_close(fp);
    }
    return fr;
}

void Mmc_Init(){
    palSetPadMode(GPIOB,13,PAL_MODE_ALTERNATE(5) |
        PAL_STM32_OSPEED_HIGHEST); //SCK

    palSetPadMode(GPIOB,12,PAL_MODE_OUTPUT_PUS
        HPULL | PAL_STM32_OSPEED_HIGHEST); //NSS
    palSetPadMode(GPIOC,2,PAL_MODE_ALTERNATE(5));
        //MISO
    palSetPadMode(GPIOC,3,PAL_MODE_ALTERNATE(5) |
        PAL_STM32_OSPEED_HIGHEST); //MOSI
    palSetPad(GPIOB, 12);

```

```

    mmcObjectInit(&MMCD1);
    mmcStart(&MMCD1, &mmccfg);
    chThdSleepMilliseconds(50);
    Mmc_Mount();
}

```

### **//mmc.h**

```

#ifndef TA_MMC_H
#define TA_MMC_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "evtimer.h"
#include "chvt.h"
#include "ff.h"
#include "ffconf.h"
#define buffer_size 16
FRESULT f_append (FIL* fp,const char* path);
FRESULT scan_files(BaseSequentialStream *chp, char *path);
void Mmc_Mount(void);
void Mmc_Unmount(void);
void Mmc_Init(void);
#endif // TA_MMC_H

```

### **//rtc.c**

```

#include "ta_rtc.h"

struct ds1307_t calendar;
static uint8_t rxbuf[DS1307_RX_DEPTH];
static uint8_t txbuf[DS1307_TX_DEPTH];

```

```

static i2cflags_t errors = 0;
uint8_t bcd2Dec ( uint8_t val )
{
    uint8_t res = ((val/16*10) + (val % 16));
    return res;
}
uint8_t dec2Bcd ( uint8_t val )
{
    uint8_t res = ((val/10*16) + (val%10));
    return res;
}
void setDs1307Date ( msg_t *status, systime_t *tmo, struct
    ds1307_t dsData )
{
    txbuf[0] = DS1307_SECONDS_REG;
    txbuf[1] = dec2Bcd( dsData.seconds );
    txbuf[2] = dec2Bcd( dsData.minutes );
    txbuf[3] = dec2Bcd( dsData.hours );
    txbuf[4] = dec2Bcd( dsData.day );
    txbuf[5] = dec2Bcd( dsData.date );
    txbuf[6] = dec2Bcd( dsData.month );
    txbuf[7] = dec2Bcd( dsData.year - 2000);

    i2cAcquireBus ( &I2CD1 );
    *status = i2cMasterTransmitTimeout ( &I2CD1,
        DS1307_ADDRESS, txbuf, DS1307_TX_DEPTH, rxbuf,
        0, *tmo );
    i2cReleaseBus ( &I2CD1 );
}

struct ds1307_t getDs1307Date ( msg_t *status, systime_t *tmo )
{
    struct ds1307_t dsData;
    txbuf[0] = DS1307_SECONDS_REG;
    i2cAcquireBus( &I2CD1 );

```

```

*status    =    i2cMasterTransmitTimeout    (    &I2CD1,
        DS1307_ADDRESS, txbuf, 1,rxbuf, 7, *tmo );
i2cReleaseBus ( &I2CD1 );
if ( *status != MSG_OK )
{
    errors = i2cGetErrors ( &I2CD1 );
}
else
{
    dsData.seconds = bcd2Dec ( rxbuf[0] & 0x7F );
    dsData.minutes = bcd2Dec ( rxbuf[1] );
    dsData.hours   = bcd2Dec ( rxbuf[2] & 0x3F );
    dsData.day     = bcd2Dec ( rxbuf[3] );
    dsData.date    = bcd2Dec ( rxbuf[4] );
    dsData.month   = bcd2Dec ( rxbuf[5] );
    dsData.year    = bcd2Dec ( rxbuf[6] ) + 2000;
}
return dsData;
}

static THD_WORKING_AREA(waRTC, 128);
static THD_FUNCTION(ThdRTC, arg) {
    (void)arg;
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    chRegSetThreadName("RTC Request");
    while (TRUE) {
        calendar = getDs1307Date ( &status, &timeOut );
        palSetPad(GPIOD, 14);    /* Red. */
        chThdSleepMilliseconds(500);
        palClearPad(GPIOD, 14); /* Red. */
        chThdSleepMilliseconds(500);
    }
}

void Ds1307_Init ( void )
{
    I2c_Init();

```



```

chThdSleepMilliseconds(500);
    palSetPadMode(GPIOD,14,PAL_MODE_OUTPUT_PUS
        HPULL);
chThdCreateStatic(waRTC, sizeof(waRTC), NORMALPRIO,
    ThdRTC, NULL);
}

```

## //rtc.h

```

#ifndef TA_RTC_H
#define TA_RTC_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "shell.h"
#include "ta_i2c.h"
#define DS1307_RX_DEPTH 7
#define DS1307_TX_DEPTH 8
#define DS1307_ADDRESS    0x68
#define DS1307_SECONDS_REG    0x00
typedef struct ds1307_t
{
    uint8_t  seconds;
    uint8_t  minutes;
    uint8_t  hours;
    uint8_t  day;
    uint8_t  date;
    uint8_t  month;
    uint16_t year;
} ds1307;
uint8_t bcd2Dec ( uint8_t val );

```

```

uint8_t dec2Bcd ( uint8_t val );
void Ds1307_Init ( void );
void setDs1307Date ( msg_t *status, systime_t *tmo, struct
    ds1307_t dsData );
struct ds1307_t getDs1307Date ( msg_t *status, systime_t *tmo );
#endif

```

**//shell.c**

```
#include "ta_shell.h"
```

```

extern uint16_t adc_co,adc_so2,adc_co2,adc_nox;
thread_t *shelltp = NULL;
extern const USBConfig usbcfg;
extern SerialUSBConfig serusbcfg;
extern struct ds1307_t calendar;
extern FATFS MMC_FS;
extern bool fs_ready;
extern uint32_t clusters;
extern FATFS *fsp;
extern uint8_t fbuff[1024];
extern FRESULT err;
SerialUSBDriver SDU1;
static void cmd_mem(BaseSequentialStream *chp, int argc, char
    *argv[]) {
    size_t n, size;
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: mem\r\n");
        return;
    }
    n = chHeapStatus(NULL, &size);
    chprintf(chp, "core free memory : %u bytes\r\n",
        chCoreGetStatusX());
    chprintf(chp, "heap fragments : %u\r\n", n);
    chprintf(chp, "heap free total : %u bytes\r\n", size);
}

```

```

static void cmd_threads(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    static const char *states[] = {CH_STATE_NAMES};
    thread_t *tp;
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: threads\r\n");
        return;
    }
    chprintf(chp, "  addr  stack prio refs  state time\r\n");
    tp = chRegFirstThread();
    do {
        chprintf(chp, "%08lx %08lx %4lu %4lu %9s\r\n",
            (uint32_t)tp, (uint32_t)tp->p_ctx.r13,
            (uint32_t)tp->p_prio, (uint32_t)(tp->p_refs - 1),
            states[tp->p_state]);
        tp = chRegNextThread(tp);
    } while (tp != NULL);
}

static void cmd_now(BaseSequentialStream *chp, int argc, char
    *argv[]) {

    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: now\r\n");
        return;
    }
    chprintf(chp, "#year = %4i\r\n",calendar.year);
    chprintf(chp, "#month = %2i\r\n",calendar.month);
    chprintf(chp, "#date = %2i\r\n",calendar.date);
    chprintf(chp, "#day = %1i\r\n",calendar.day);
    chprintf(chp, "#hour = %2i\r\n",calendar.hours);
    chprintf(chp, "#minute= %2i\r\n",calendar.minutes);
    chprintf(chp, "#second= %2i\r\n",calendar.seconds);
}

```

```

static void cmd_settime(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    if (argc != 3) {
        chprintf(chp, "Usage: settime sec min hr\r\n");
        return;
    }
    calendar.seconds = atoi(argv[0]);
    calendar.minutes = atoi(argv[1]);
    calendar.hours = atoi(argv[2]);
    setDs1307Date( &status, &timeOut, calendar);
    chprintf(chp, "time was set\r\n");
}

static void cmd_setdate(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    if (argc != 3) {
        chprintf(chp, "Usage: setdate date month year\r\n");
        return;
    }
    calendar.date = atoi(argv[0]);
    calendar.month = atoi(argv[1]);
    calendar.year = atoi(argv[2]);
    setDs1307Date( &status, &timeOut, calendar);
    chprintf(chp, "date was set\r\n");
}

static void cmd_mmctree(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmctree\r\n");
        return;
    }
}

```

```

    if (!fs_ready) {
        chprintf(chp, "File System not mounted\r\n");
        return;
    }
    err = f_getfree("/", &clusters, &fsp);
    if (err != FR_OK) {
        chprintf(chp, "FS: f_getfree() failed (%i)\r\n",err);
        return;
    }
    chprintf(chp,"FS: %lu free clusters, %lu sectors per cluster,
        %lu          bytes          free\r\n",clusters,
        (uint32_t)MMC_FS.csize,clusters          *
        (uint32_t)MMC_FS.csize                    *
        (uint32_t)MMC_SECTOR_SIZE);
    fbuff[0] = 0;
    scan_files(chp, (char *)fbuff);
}
static void cmd_mmctest(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmctest\r\n");
        return;
    }
    if (!fs_ready) {
        chprintf(chp, "File System not mounted\r\n");
        return;
    }
    FIL FDLogFile;
    memset(&FDLogFile, 0, sizeof(FIL));
    FRESULT err_file;
    UINT bw;
    char buffer[buffer_size];
    err_file = f_open(&FDLogFile, "Test.txt", FA_WRITE |
        FA_OPEN_ALWAYS);

```

```

if (err_file == FR_OK || err_file == FR_EXIST){
    err_file = f_lseek(&FDLogFile, f_size(&FDLogFile));
    if(err_file == FR_OK){
        chsnprintf(buffer,buffer_size,"Aku Jomblo!!!\n\r");
        f_write(&FDLogFile, buffer, strlen(buffer), &bw);
        f_close(&FDLogFile);
        chprintf(chp, "Some text written\r\n");
        return;
    }else{
        chprintf(chp, "Failed to seek file\r\n");
        return;
    }
}else{
    chprintf(chp, "Cannot Write file\r\n");
    return;
}
}

static void cmd_mmcadc(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmcadc\r\n");
        return;
    }
    if (!fs_ready) {
        chprintf(chp, "File System not mounted\r\n");
        return;
    }
    Tulis_Adc();
}

static void cmd_dataadc(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: dataadc\r\n");
        return;
    }
}

```

```

    }
    {
        chprintf(chp, "adc_co = %4i\r\n",adc_co);
        chprintf(chp, "adc_so2= %4i\r\n",adc_so2);
        chprintf(chp, "adc_co2= %4i\r\n",adc_co2);
        chprintf(chp, "adc_nox= %4i\r\n",adc_nox);
    }
}

static void cmd_testsms(BaseSequentialStream *chp, int argc,
                        char *argv[]) {
    (void) argv;
    if (argc > 0) {
        chprintf(chp, "Usage: testsms\r\n");
        return;
    }
    Sms_Test();
}

static const ShellCommand commands[] = {
    {"mem", cmd_mem},
    {"threads", cmd_threads},
    {"now", cmd_now},
    {"settime", cmd_settime},
    {"setdate", cmd_setdate},
    {"mmctree", cmd_mmctree},
    {"mmctest", cmd_mmctest},
    {"mmcadc", cmd_mmcadc },
    {"dataadc", cmd_dataadc},
    {"testsms", cmd_testsms},
    {NULL, NULL}
};

static const ShellConfig shell_cfg = {
    (BaseSequentialStream *)&SDU1,
    commands
};

void Shell_Init(void){
    sduObjectInit(&SDU1);

```

```

    sduStart(&SDU1, &serusbcfg);
    usbDisconnectBus(serusbcfg.usbp);
    chThdSleepMilliseconds(1000);
    usbStart(serusbcfg.usbp, &usbcfg);
    usbConnectBus(serusbcfg.usbp);
    shellInit();
}
void Shell_Run(void){
    if (!shelltp && (SDU1.config->usbp->state ==
        USB_ACTIVE))
        shelltp = shellCreate(&shell_cfg, SHELL_WA_SIZE,
            NORMALPRIO);
    else if (chThdTerminatedX(shelltp)) {
        chThdRelease(shelltp); /* Recovers memory of the previous
            shell. */
        shelltp = NULL; /* Triggers spawning of a new shell.
            */
    }
    chThdSleepMilliseconds(1000);
}

```

## //shell.h

```

#ifndef TA_SHELL_H
#define TA_SHELL_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "shell.h"
#include "ta_usbcfg.h"
#include "ta_utama.h"

```



```

#include "ta_mmc.h"
#include "ta_uart.h"
#define                                SHELL_WA_SIZE
    THD_WORKING_AREA_SIZE(4096)
#define                                TEST_WA_SIZE
    THD_WORKING_AREA_SIZE(256)

void Shell_Init(void);
void Shell_Run(void);
#endif // TA_SHELL_H

```

**//uart.c**

```

#include "ta_uart.h"

extern struct ds1307_t calendar;
extern uint16_t adc_co,adc_so2,adc_co2,adc_nox;
extern float v_co,v_so2,v_co2,v_nox;
void Uart_Init(void){
    palSetPadMode(GPIOB,11,PAL_MODE_ALTERNATE(7));
    palSetPadMode(GPIOB,10,PAL_MODE_ALTERNATE(7));
    sdStart(&SD3,NULL);
}
void Sms_Text(void){
    Uart_Init();
    chThdSleepMilliseconds(500);
}
void Sms_Test(void){
    chprintf((BaseSequentialStream *)&SD3,"AT+CMGF=1\n");
    chThdSleepMilliseconds(100);
    chprintf((BaseSequentialStream *)&SD3,"AT+CMGS=\"");
    chprintf((BaseSequentialStream *)&SD3,"+6282244105564");
    chprintf((BaseSequentialStream *)&SD3,"\\\"\\n");
    chThdSleepMilliseconds(100);
    chprintf((BaseSequentialStream *)&SD3,"Hasil Monitoring
        Gas pada\\n");
}

```

```

chprintf((BaseSequentialStream *)&SD3,"Tanggal   =%2i-%2i-
    %4i                                     pukul
    %2i:%2i\n",calendar.date,calendar.month,calendar.year,cal
    endar.hours,calendar.minutes);
chprintf((BaseSequentialStream *)&SD3,"co       =%4i
    %7.1f\n",adc_co,v_co);
chprintf((BaseSequentialStream *)&SD3,"so2=%4i
    %7.1f\n",adc_so2,v_so2);
chprintf((BaseSequentialStream *)&SD3,"co2=%4i
    %7.1f\n",adc_co2,v_co2);
chprintf((BaseSequentialStream *)&SD3,"nox=%4i
    %7.1f\n",adc_nox,v_nox) ;
chThdSleepMilliseconds(100);

chSequentialStreamPut((BaseSequentialStream
    *)&SD3,0x1A);
chSequentialStreamPut((BaseSequentialStream
    *)&SD3,0x0D);
chSequentialStreamPut((BaseSequentialStream
    *)&SD3,0x0A);
chprintf((BaseSequentialStream *)&SD3,"\n");
}

```

## //uart.h

```

#ifndef TA_UART_H
#define TA_UART_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "ta_rtc.h"

```

```
#include "ta_utama.h"
```

```
void Uart_Init(void);  
void Sms_Text(void);  
void Sms_Test(void);  
#endif // TA_UART_H
```

```
//utama.c
```

```
#include "ta_utama.h"
```

```
extern LcdStream myLCD;  
extern adcsample_t adc0,adc1,adc2,adc3;  
uint16_t adc_co,adc_so2,adc_co2,adc_nox;  
float v_co,v_so2,v_co2,v_nox,v_in,R_S;  
extern struct ds1307_t calendar;  
extern SerialUSBDriver SDU1;  
extern FATFS MMC_FS;  
uint8_t udhkirim=0;  
static THD_WORKING_AREA(waBlink, 128);  
static THD_FUNCTION(Blink, arg) {  
  
    (void)arg;  
    chRegSetThreadName("Blinker");  
    while (TRUE) {  
        palSetPad(GPIOD, 13);    /* Orange. */  
        chThdSleepMilliseconds(500);  
        palClearPad(GPIOD, 13);  /* Orange. */  
        chThdSleepMilliseconds(500);  
    }  
}  
  
static THD_WORKING_AREA(waADCLCD, 128);  
static THD_FUNCTION(ADCLCD, arg) {  
  
    (void)arg;  
    chRegSetThreadName("ADC LCD");  
    while (TRUE) {
```

```

    Hasil_Adc();
}
}
static THD_WORKING_AREA(waRECORD, 1024);
static THD_FUNCTION(RECORD, arg) {

    (void)arg;
    chRegSetThreadName("RECORD");
    while (TRUE) {
        Tulis_Adc();
        palClearPad(GPIOD, 15);
        chThdSleepMilliseconds(5000);
        if ((v_co >= 25) || (v_so2 >= 20) || (v_co2 >= 600) || (v_nox
            >= 4)){
            if(udhkirim==0){
                Sms_Test();
                udhkirim=1;
            }
        }
        else{
            udhkirim=0;
        }
    }
}

void Run_Init(void){
    chThdCreateStatic(waADCLCD,          sizeof(waADCLCD),
        NORMALPRIO,          ADCLCD,          NULL);
    palSetPadMode(GPIOD,13,PAL_MODE_OUTPUT_PUS
        HPULL);
    palSetPadMode(GPIOD,15,PAL_MODE_OUTPUT_PUS
        HPULL);
    chThdCreateStatic(waBlink, sizeof(waBlink), NORMALPRIO,
        Blink, NULL);
    chThdSleepMilliseconds(1000);
    chThdCreateStatic(waRECORD,          sizeof(waRECORD),
        NORMALPRIO, RECORD, NULL);

```

```

}
void Hasil_Adc(void){
    adc_co =adc0;
    adc_so2=adc1;
    adc_co2=adc2;
    adc_nox=adc3;

    v_co = (0.2279*adc_co)-32.492;
    v_so2 = (0.0018*adc_so2)-1.205;
    v_co2 = ((-0.2446*adc_co2)+926.94);
    v_nox = ((adc3* 9.9)/4095)*0.1;

    Lcd_Cursor(0,0);
    chprintf((BaseSequentialStream *)&myLCD,"co    =%7.1f
    ppm",v_co);
    Lcd_Cursor(0,1);
    chprintf((BaseSequentialStream *)&myLCD,"so2=%7.3f
    ppm",v_so2);
    Lcd_Cursor(0,2);
    chprintf((BaseSequentialStream *)&myLCD,"co2=%7.1f
    ppm",v_co2);
    Lcd_Cursor(0,3);
    chprintf((BaseSequentialStream *)&myLCD,"nox=%7.2f
    ppm",v_nox) ;
}

```

```

void Tulis_Adc(void){
    FIL FDLogFile;
    memset(&FDLogFile, 0, sizeof(FIL));
    FRESULT err_file;
    UINT bw;
    char buffer[64];

    palSetPad(GPIOD, 15);
}

```

```

err_file = f_open(&FDLogFile, "Data Monitoring.csv",
    FA_WRITE | FA_OPEN_ALWAYS );
if (err_file == FR_OK || err_file == FR_EXIST){
    err_file = f_lseek(&FDLogFile, f_size(&FDLogFile));
    if(err_file == FR_OK){

        chsnprintf(buffer,64,"%2i-%2i-
        %4i;%2i:%2i;%4i;%4i;%4i;%4i\r\n",calendar.date,calendar
        .month,calendar.year,calendar.hours,calendar.minutes,adc_
        co,adc_so2,adc_co2,adc_nox);
        f_write(&FDLogFile, buffer, strlen(buffer), &bw);
        f_close(&FDLogFile);
        chprintf((BaseSequentialStream *)&SDU1, "Some text
        written\r\n");
        return;
    }else{
        chprintf((BaseSequentialStream *)&SDU1, "Failed to
        seek file\r\n");
        return;
    }
}else{
    chprintf((BaseSequentialStream *)&SDU1, "Cannot Write
    file\r\n");
    return;
}
}

```

**//utama.h**

```

#ifndef TA_UTAMA_H
#define TA_UTAMA_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"

```

```
#include "chprintf.h"  
#include "memstreams.h"  
#include "chstreams.h"  
#include "ta_adc.h"  
#include "ta_lcd.h"  
#include "ta_rtc.h"  
#include "ta_mmc.h"  
#include "ta_uart.h"
```

```
void Run_Init(void);  
void Hasil_Adc(void);  
void Tulis_Adc(void);
```

```
#endif // TA_UTAMA_H
```

## LAMPIRAN B

### (DATA SHEET STM32F4 DISCOVERY)



UM1472  
User manual

Discovery kit with STM32F407VG MCU

### Introduction

The STM32F4DISCOVERY Discovery kit allows users to easily develop applications with the STM32F407VG high performance microcontroller with the ARM® Cortex®-M4 32-bit core. It includes everything required either for beginners or for experienced users to get quickly started.

Based on STM32F407VG, it includes an ST-LINK/V2 or ST-LINK/V2-A embedded debug tool, two ST-MEMS digital accelerometers, a digital microphone, one audio DAC with integrated class D speaker driver, LEDs, push buttons and a USB OTG micro-AB connector. To expand the functionality of the STM32F4DISCOVERY Discovery kit with the Ethernet connectivity, LCD display and more, visit the [www.st.com/stm32f4dis-expansion](http://www.st.com/stm32f4dis-expansion) webpage. The STM32F4DISCOVERY Discovery kit comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube package, as well as a direct access to the ARM® mbed Enabled™ on-line resources at <http://mbed.org>.

Figure 1. STM32F4DISCOVERY



1. Picture is not contractual.





## 1 Features

The STM32F4DISCOVERY offers the following features:

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package
- On-board ST-LINK/V2 on STM32F4DISCOVERY or ST-LINK/V2-A on STM32F407G-DISC1
- ARM® mbed Enabled™ (<http://mbed.org>) with ST-LINK/V2-A only
- USB ST-LINK with re-enumeration capability and three different interfaces:
  - Virtual COM port (with ST-LINK/V2-A only)
  - Mass storage (with ST-LINK/V2-A only)
  - Debug port
- Board power supply:
  - Through USB bus
  - External power sources:  
3 V and 5 V
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02 ST MEMS audio sensor omni-directional digital microphone
- CS43L22 audio DAC with integrated class D speaker driver
- Eight LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power on
  - Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  - 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current
- Two push buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing
- Comprehensive free software including a variety of examples, part of the STM32CubeF4 package or STSW-STM32068 for legacy standard library usage

## 2 Product marking

Tools marked as "ES" or "E" are not yet qualified and as such, they may be used only for evaluation purposes. ST shall not be liable for any consequences related with other ways of use of such non-qualified tools, for example, as reference design or for production.

Examples of location of "E" or "ES" marking:

- On target STM32 microcontroller part mounted on the board (for illustration, refer to section "Package information" of a STM32 datasheet at [www.st.com](http://www.st.com)).
- Next to the evaluation tool ordering part number, as a label stuck or a silk-screen printed on the board.

## 3 Ordering information

To order the Discovery kit for the STM32F407 line of microcontrollers, refer to [Table 1](#).

Table 1. List of the order codes

Order code	ST-LINK version
STM32F4DISCOVERY	ST-LINK/V2
STM32F407G-DISC1	ST-LINK/V2-A (mbed Enabled)

## 4 Conventions

[Table 2](#) provides the definition of some conventions used in the present document.

Table 2. ON/OFF conventions

Convention	Definition
Jumper JP1 ON	Jumper fitted
Jumper JP1 OFF	Jumper not fitted
Solder bridge SBx ON	SBx connections closed by solder
Solder bridge SBx OFF	SBx connections left open

## 5 Quick start

The STM32F4DISCOVERY is a low-cost and easy-to-use development kit to quickly evaluate and start a development with an STM32F407VG high-performance microcontroller.

Before installing and using the product, accept the Evaluation Product License Agreement from the [www.st.com/stm32f4-discovery](http://www.st.com/stm32f4-discovery) webpage.

For more information on the STM32F4DISCOVERY and for demonstration software, visit the [www.st.com/stm32f4-discovery](http://www.st.com/stm32f4-discovery) webpage.

### 5.1 Getting started

Follow the sequence below to configure the STM32F4DISCOVERY board and launch the DISCOVER application:

1. Check jumper position on the board, JP1 on, CN3 on (DISCOVERY selected).
2. Connect the STM32F4DISCOVERY board to a PC with a USB cable 'type A to mini-B' through USB connector CN1 to power the board. Red LED LD2 (PWR) then lights up.
3. Four LEDs between B1 and B2 buttons are blinking.
4. Press user button B1 to enable the ST MEMS sensor, move the board and observe the four LEDs blinking according to the motion direction and speed. (If a second USB cable 'type A to micro-B' is connected between PC and CN5 connector, then the board is recognized as standard mouse and its motion will also control the PC cursor).
5. To study or modify the DISCOVER project related to this demonstration, visit the [www.st.com/stm32f4-discovery](http://www.st.com/stm32f4-discovery) webpage and follow the tutorial.
6. Discover the STM32F407VG features, download and execute programs proposed in the list of projects.
7. Develop the application using available examples.

### 5.2 System requirements

- Windows® OS (XP, 7, 8 and 10), Linux® 64-bit or macOS™
- USB type A to Mini-B cable.

### 5.3 Development toolchains supported

- Keil® MDK-ARM<sup>(a)</sup>
- IAR™ EWARM<sup>(a)</sup>
- GCC-based IDEs including free SW4STM32 from AC6
- ARM® mbed Enabled™ online

---

a. On Windows® only.

## 6 Hardware and layout

The STM32F4DISCOVERY is designed around the STM32F407VGT6 microcontroller in a 100-pin LQFP package.

Figure 2 illustrates the connections between the STM32F407VGT6 and its peripherals (ST-LINK/V2 or ST-LINK/V2-A, push buttons, LEDs, Audio DAC, USB, ST-MEMS accelerometer and microphone, and connectors).

Figure 3 and Figure 4 help users to locate these features on the STM32F4DISCOVERY board.

Figure 2. Hardware block diagram

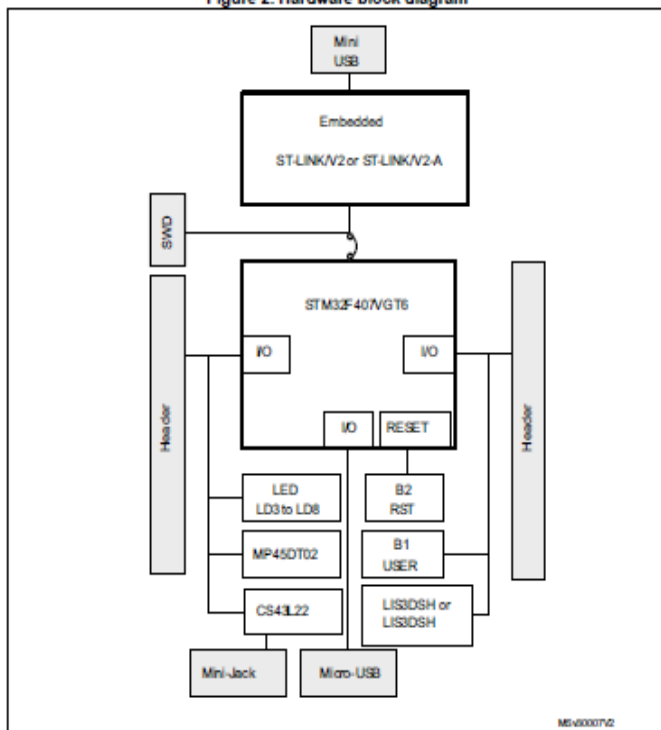
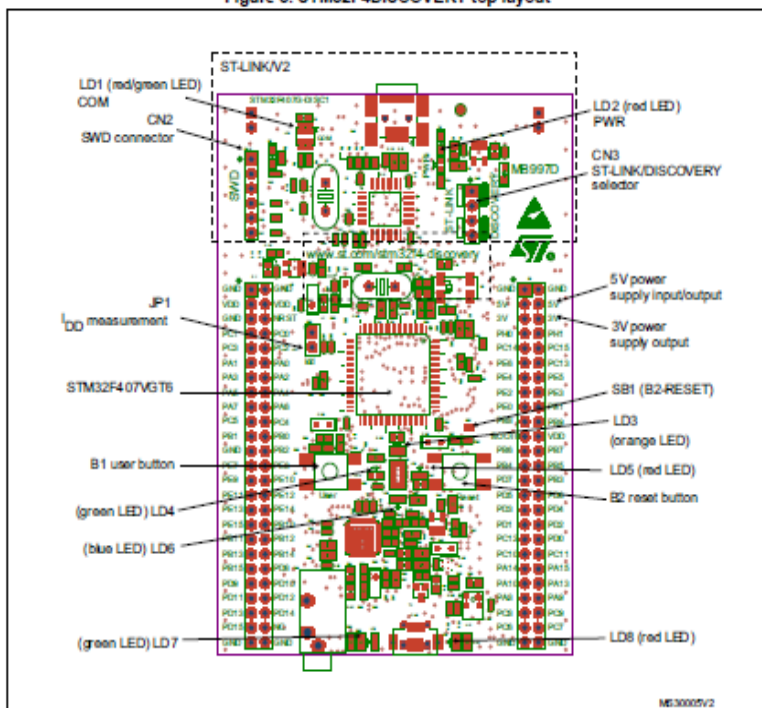
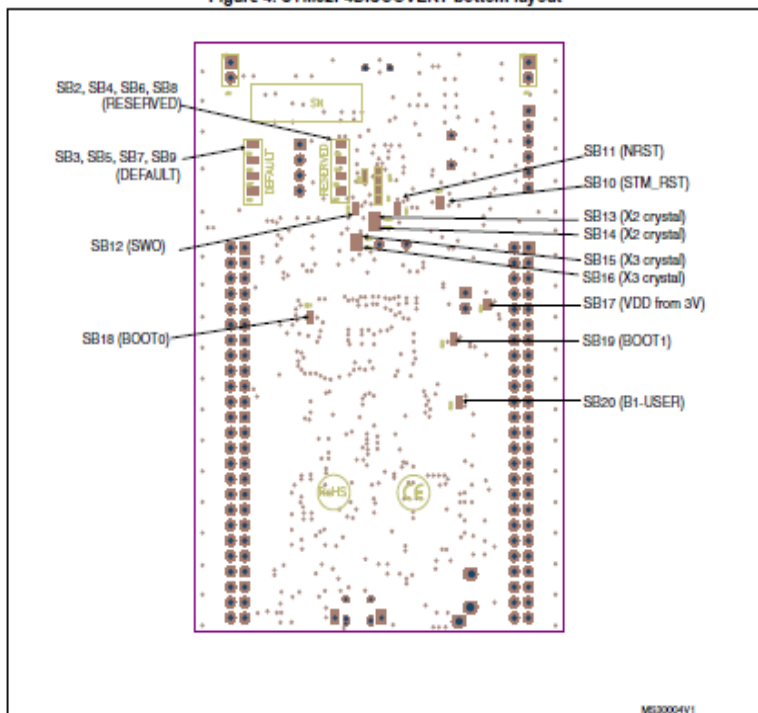


Figure 3. STM32F4DISCOVERY top layout



Note: Pin 1 of CN2, CN3, JP1, P1 and P2 connectors are identified by a red square.

Figure 4. STM32F4DISCOVERY bottom layout



## 6.1 Embedded ST-LINK/V2 (or V2-A)

ST-LINK/V2 on STM32F4DISCOVERY or ST-LINK/V2-A on STM32F407G-DISC1 is an embedded tool for programming and debugging.

The embedded ST-LINK/V2 (or V2-A) supports only SWD for STM32 devices. For information about debugging and programming features refer to *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*, UM1075 User manual, which describes in details all the ST-LINK/V2 features.

The changes on ST-LINK/V2-A versus ST-LINK/V2 version are listed below.

New features supported on ST-LINK/V2-A:

- Virtual COM port interface on USB (see [Section 6.1.3: ST-LINK/V2-A VCP configuration](#))
- Mass storage interface on USB



current should be control below 1pA.

### Specifications :

Symbol	Parameter Name	Technical	Remarks
$V_H$	Heating Voltage	$6.0 \pm 0.1 \text{ V}$	AC or DC
$R_H$	Heating Resistor	$30.0 \pm 5\% \Omega$	Room Temperature
$I_H$	Heating Current	@200mA	
$P_H$	Heating Power	@1200mW	
$T_{ao}$	Operating Temperature	$-20 \dots +50^\circ\text{C}$	
$T_{as}$	Storage Temperature	$-20 \dots +70^\circ\text{C}$	
□ EMF	Output	30—50mV	350—1000ppm CO <sub>2</sub>

### Sensitivity :

Figure 2 Shows gas sensor sensitivity curve. :

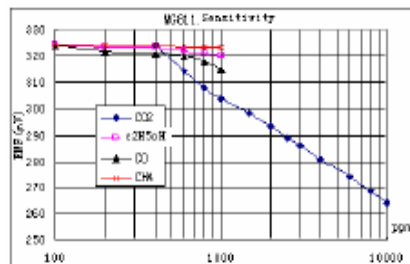
Conditions:

Tem :  $28^\circ\text{C}$

RH : 65%

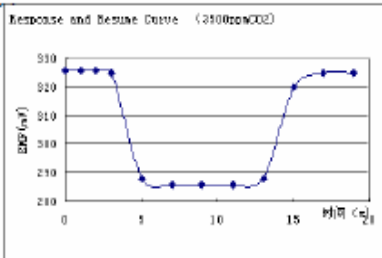
Oxygen : 21%

EMF: sensor EMF under different gas and concentration.

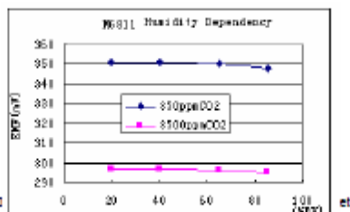
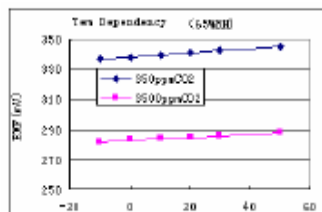


### Response and Resume Characteristic :

Figure 3 shows Solid electrolyte sensor response and resume characteristics.



### Temperature and Humidity Dependency :





# CO2 Sensor SKU:SEN0159

From Robot Wiki

## Contents

- 1 Introduction
- 2 Specification
- 3 Connecting Diagram
- 4 Tutorial
- 5 Sample code
  - 5.1 Sample code1
  - 5.2 Sample code2



CO2 Sensor (Arduino compatible) SKU:SEN0159

## Introduction

"Greenhouse Effect" is melting the iceberg every minute. By knowing the exact concentration of CO<sub>2</sub>, we can do something to reduce the CO<sub>2</sub> and to protect our earth. For that reason, a HQ CO<sub>2</sub> sensor is designed by DFRobot engineer. This is the first CO<sub>2</sub> sensor in OSHW market. The output voltage of the module falls as the concentration of the CO<sub>2</sub> increases. The potentiometer onboard is designed to set the threshold of voltage. As long as the CO<sub>2</sub> concentration is high enough (voltage is lower than threshold), a digital signal (ON/OFF) will be released.

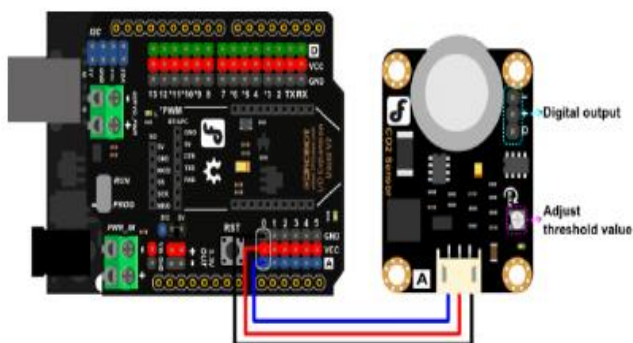
- It has MG-811 sensor module onboard which is highly sensitive to CO<sub>2</sub> and less sensitive to alcohol and CO. Low humidity&temperature dependency. All components have industrial quality which means stability and reproducibility.
- Onboard heating circuit brings the best temperature for sensor to function. 5V power input will be boosted to 6V for heating.
- This sensor has an onboard conditioning circuit for amplifying output signal.

Notice: 7~12V is necessary to power the microcontroller when using the CO<sub>2</sub> Sensor

## Specification

- Operating voltage: 5V
- Interface: Analog
- One digital output
- High quality connector
- Immersion gold surface
- Onboard heating circuit
- Size: 32x42mm

## Connecting Diagram



## Tutorial

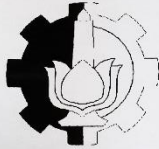
How to use this module?

It is very easy.

You need to set potentiometer onboard to the threshold value. Just make the red led turn off. With the CO2 concentration is enough high to make the sensor output voltage higher than threshold value, the led will be turned on. If you connect a buzzer to the module(right side), you will hear the alarm.



## LAMPIRAN D (LAPORAN HASIL PENGAMBILAN DATA)



LABORATORIUM PENCEMARAN UDARA DAN PERUBAHAN IKLIM  
DEPARTEMEN TEKNIK LINGKUNGAN  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

KAMPUS ITS SUKOLILO SURABAYA  
TELPON (031)5948886, FAX. (031)5928397

### DATA KALIBRASI ALAT DETEKTOR PENCEMAR UDARA

Pengirim : Syahril Arisdianto / 2414031017  
Jenis Pencemar : Karbon Dioksida ( CO<sub>2</sub> )

Pengukuran	A D C	Standard ( ppm )
1	2155	396
2	2117	415
3	1963	448
4	1848	471
5	1752	499
6	1560	546
7	1445	570
8	1349	598
9	1214	636
10	1138	645

Surabaya, 20 Juli 2017  
Kepala Laboratorium Pencemaran Udara dan Perubahan Iklim  
Departemen Teknik Lingkungan FTSP-ITS



Dr.Eng. Ari Dharma Syari, ST.MEPM  
NIP : 1982-01-19-200501 1 001

## BIODATA PENULIS



Nama lengkap penulis Syahril Arisdianta yang dilahirkan di Sumenep pada tanggal 09 Januari 1996 dari ayah bernama Syaifuddin dan ibu Sri Hastuti. Penulis merupakan anak kedua dari dua bersaudara. Saat ini penulis tinggal di Sumenep. Pada tahun 2008, penulis menyelesaikan pendidikan tingkat dasar di SDN Pajagalan 2 Sumenep. Pada tahun 2011 penulis menyelesaikan pendidikan tingkat menengah pertama di SMPN 1 Sumenep. Tahun 2014 berhasil menyelesaikan pendidikan tingkat menengah atas di SMAN 1 Sumenep. Dan pada tahun 2017 ini, penulis mampu menyelesaikan gelar ahli madya di Program Studi DIII Teknik Instrumentasi, Departemen Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya. Penulis berhasil menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Monitoring Konsentrasi Gas Karbon Dioksida (CO<sub>2</sub>) Sebagai Emisi Gas Buang Menggunakan Sensor MG 811 Berbasis Mikrokontroler STM32F4 Discovery”**. Bagi pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai Tugas Akhir ini maka dapat menghubungi penulis melalui email [syahril.arisdianta@gmail.com](mailto:syahril.arisdianta@gmail.com).